NSRP 0256

COMPUTER-ASSISTED METHODOLOGY FOR THE DETERMINATION OF THE OPTIMAL NUMBER AND LOCATION OF TOOL SHEDS

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
JUL 1986	**N/A**	-

4. TITLE AND SUBTITLE	
Computer-Assisted Methodology for the Determination of the Optimal Number and Location of Tool Sheds	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Naval Surface Warfare Center CD Code 2230 - Design Integration Tools Building 192 Room 128 9500 MacArthur Blvd Bethesda, MD 20817-5700**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**SAR**	**128**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

THE NATIONAL SHIPBUILDING RESEARCH PROGRAM

FINAL REPORT FOR TASK EC-26

Computer-Assisted Methodology for the Determination of the Optimal Number and Location of Tool Sheds

Dr. Scott Christian Iverson
Associate Professor

University of Washington
Department of Mechanical Engineering
FU-10
Seattle, Washington 98195
(206) 543-5398, (206) 362-7745

THE NATIONAL SHIPBUILDING RESEARCH PROGRAM

Task EC-26
Computer-Assisted Methodology
for the Determination
of the Optimal Number
and Location of Tool Sheds

CONDUCTED AT:
Mechanical Enqineerinq Department FU-10
Industrial Enqineerinq Proqram
University of Washington
Seattle, Washinqton 98195

January 1986 throuqh July 1986

FOR :

Bath Iron Works Corporation
700 Washinqton St.
Bath, Maine 04530

The Society of Naval
Architects and Marine Enqineers
Ship Production Committee
Sp-8 Panel on Industrial Enqineerinq

The U.S. Department of Transportation
Maritime Administration

This project is managed and cost-shared by Bath Iron Works Corporation for the National Shipbuilding Research Program. The program is a cooperative effort of the Maritime Administration’s Office of Advanced Ship Development, the U.S. Navy, the U.S. shipbuilding industry, and selected academic institutions.

Executive Summary

This project was performed by the University of Washington to provide computer assistance for choosinq optimal locations for toolrooms in shipyards. The tool used to accomplish this task is a computer proqram entitled Computer-Assisted Toolroom Desiqn (CATD). The CATD program is executable on the IBM-AT and the IBM-PC with an 8087 co-processor chip. Its operation and maintenance is outlined in the user and technical manual enclosed in the report.

The CATD proqram provides the user with the ability to analyze a shipyard with respect to the utilization of its resources in reqards to toolroom locations. By determining optimal toolroom locations, the user gains an insiqht of the system and the systems possible improvements by varyinq the locations. Futhermore, proposed toolrooms can be tested in the proqram to determine their economic feasibility for expansion. An application of the program was performed at Inqall's Shipyard.

The program will be most applicable at shipyards that have limited resources in workforce? toolrooms and equipment. Ten to fifteen percent cost savinqs based on the craftsmen unproductive time spent queueing and travelinq, can be estimated for shipyards from the preliminary analysis.

Table of Contents

Section 1: Report to Management Page

Introduction......................................1.1

Description of Documentation..................1.1-1.2
User Manual
Technical Manual
The CATD Program..............................1.2-1.3

Applications..................................1.3-1.5
Ingall's Shipyard
Generic Model
Potential Expansions..............................1.5

Other Applications............................1.5-1.7

Appendix.....................................1.8-1.24

Section 2: User Manual

Introduction......................................2.1

Operating the Computer........................2.1-2.5
How To Get Started
How To Get A Print-Out
Running the Program..........................2.5-2.24
Entering Data
Obtaining Coordinates
Tutorial
Interpreting the Results.........................2.24

Section 3: Technical Manual
Introduction......................................3.1

Design Process................................3.2-3.4
Requirements
Design
Code
Correcting Errors.................................3.4

How To Improve or Change A Feature............3.4-3.5

Data..3.5-3.7
Financial Information
Travel Information
Worker Information
Queueing Information
Appendix.....................................3.8-3.74

REPORT To MANAGEMENT

1.1 Introduction

CATD is a computer program written in Fortran77 that runs on the IBM-AT and is accessible on the IBM-PC with a 8087 Co-Processor Chip. It is designed to provide computer assistance for choosinq optimal locations of toolrooms in a shipyard. CATD proqram provides a solid basis for a continued research system and can be used with little difficulty. Documentation for the CATD proqram includes two manuals: (1) User Manual (2) Technical Manual.

1.2 Description of Documentation

1.2.1 User manual

The user manua1 describes the operations of the CATD proqram. Complete procedures for usinq the personal computer is subsumed, where the manual assumes that the indiviual operating the system has little experience prior to usinq the computer. Futhermore, its intended audience is an individual who will be using the tool by entering the necessary data to qet statistics about the toolroorm, and their optimal locations.

The manua1 provides a brief tutorial that walks the user throuqh the program and indicates how the data is entered. In addition, the proqram checks for mistyped data.

1.2.2 Technical Manual

The technical manua1 provides the necessary information for maintaining and upgradinq the CATD proqram. It was written for an individual responsible for maintaining the software at a shipyard. The individual should have experience in desiqninq modular proqrams and know Fortran77. The appendices contain the data flow diaqrams, data dictionary, process specifications structured charts and source listinq that describe CATD proqram. The desiqn of the proqram can be referred to and is followed by a structured analysis and structured desiqn methodology as described in Structured Analysis and System Specification by Tom de Marco (1) and The Practical Guide to Structured Systems Desiqn by Meilir Page-Jones (2). The manual describes the desiqninq process in the book and recommends the procedures for upqradinq the CATD proqram.

1.3 The CATD PR0GRAM

CATD proqram provides a user with the ability to evaluate toolroom locations for present and future uses. The proqram characterizes the toolroom environment by analyzinq the traffic of workers in the toolroom. The model l asumes that the workers who are served will randomly arrive at the toolroom; therefore, a mathematical servinq or queueinq model can be applied in this tuation. The cost of serving the toolroom is determined as a function of the productivity of the craftsmen. Their work efficiency is decreased throuqh unproductive time spent in travelinq to and queueing at the toolrooms. When the work areas of the craftsmen is specified, the CATD proqram determines the optimal location of the toolroom. Hence, the proqram is able to postulate the efficiency with which the system is currently operating and the level to which it can be improved by varying the toolroom locations.

The program also exercises flexibility by estimating the consequences or benefits of a proposed decision. For example, it allows the user to evaluate the feasibility of an additional or future toolroom, without actually implementing. In this sense, the model can be modified superficially in order to determine how the system would respond to chanqes in the number of toolrooms.

The required input for the CATD program is obtained fairly easily. An actual illustration of the program's capabilities was tested at Inqall's Shipyard.

1.4 Applications

1.4.1 Inqall's Shipyard

The data gathered at the Inqall's Shipyard consisted of four phases: financial travel, workers, and queueinq information. The financial data consisted of the buildinq and inventory cost for each toolroom, plus the averaqe worker's waqe. The travel information includes the distance between the toolroom and service areas, and the walkinq rate of a normal worker . Lastly, the worker and queueing data can be tabulated simultaneously by the attendant who records the number of workers beinq serviced in the toolroom and the time taken to service them. The technical manual provides a minimum data listinq and its mean for acquisition.

By inputtinq the data into the CATD proqram, it was determined that shipyards could anticipate a considerable savinqs by optimizing the locations of their toolrooms. The user’s manual tutorial section has the print-out of the session. For the existinq toolroom, Fabric Shop Toolroom, a cost reduction of $62.90/day was estimated. Feedback was attained at the Inqall’s Shipbuilding while demonstrateinq the use of the proqram to their Supervisor of Tool Control, Section Manaqer of Operations Data Control and various Industrial Enqineerinq staff. There existed a concensus that the tool could provide a savinqs brouqht about by increased productivity of the craftsmen and that it is useful as a decision-makinq tool in providinq evidence to support the implementation of chanqes in the location of the toolrooms.

1.4.2 Generic Model

The CATD proqram proved to be successful in the Inqall’s Shipyard, and can be effective when applied to other shipyards. The program's potential will depend largely on the shipyard setting and its operating characteristics.

Shipyard environments that have limited available resources will benefit most from the model, where limitinq resources include workforce, toolrooms, and equipment. Shipyards that are subjected to a confined number of toolrooms in proportion to its needs will find the CATD proqram can determine optimal locations for the toolroom, and aid in decision-makinq of additional toolrooms. An estimated ten to fifteen percent of the craftsmen's unproductive time is the anticipated cost savings based on the preliminary analysis. Shipyards that have a sufficient number of toolrooms that are centrally located within the service areas will find the model less useful . In addition, shipyards who distribute the tools permanently at the service areas will find the model unapplicable.

1.5 Potential Expansion

The CATD program provides a computer system that can be integrated into the shipyards. Personal computers ara powerful enouqh for supporting complicated simulation applications and expert systems. The algorthim used in determininq toolroom locations by calculating centroids of service areas can further be enhanced with a continuing deqree of sophistication. Appendix A includes a listinq of readinq with applications to further simulations that can compliment the CATD proqram.

1.6 Other Applications

The followinq section discusses research and computer applications in the shipbuilding industries and related industries.

1.6.1 Shipbuilding Industry

Computers were first used in ship desiqn. Development in this field over the past two decades has allowed a full structural analysis to be performed on marine vehicles. Computer graphics has played a major role in updatinq the traditional methods.

In consideration of preliminary desiqn, the computer is used primarily for producing a range of feasible designs. However, problems lie in the selection of criteria for evaluating the design. Advanced technology provides the designer with computer-aided-design and manufacturing (CAD,CAM) for revising final drawings used for the production process. Preparation-effort and draftinq time has been greatly reduced by these methods. In addition, Hull Surface Generation has developed into three major approaches: Computerization of the Graphical Method, Mathematical Fairing, Surface Generation. Each approach is essentially a fittinq process within specified parameters.

Computer software is presently, being developed for facilitating decision-makinq and the control of production. The packages are yet to be fully satisfactory. Many material billing, inventory control and purchasing software have been developed, but its practicalities will depend on the qoal of the company. It is clear that there is still further development in the application of computers. Some examples of computer integration and research for shipyards are found in Appendix B.

1.6.2 Other Industries

The computer is fully accepted as a tool for achieving efficiency and cost-effectiveness in a larqe number of industries such as automotive, aerospace, and manufacturing. These industries are analogical to the production process in shipbuilding for these industries produce large scale assembled products. The car assembly line is a good example of robot use in the manufacturing industry. Robots have not yet come into widespread general use in the shipbuilding industry because of problems with the large geometry of the work pieces. Greater automation in the shipyard is expected to increase and develop. In Appendix B, there are articles supplemented with brief descriptions to gain an overview of what other industries are doing to increase production through research into computer applications.

Appendix A

Potential Expansions

Diesslin, R.L. "A Survey of CAD/CAM Technology Applications in the U.S. Shipbuilding Industry." National Shipbuilding Research Program report 0188, Department of Transportation,Washinqton,DC

Summary:

This paper presents a comprehensive survey of CAD/CAM equipment installed and planned for installation in U.S. shipyards. The main emphasis of the paper was ratinq the various applications for quality of software and effectiveness of applications in executing the assigned tasks. Future developments sought by the shipyards were described and problems with acheivinq these goals were discussed.

Example:

none given

Application:

This survey gives a good overview of the installed computer base in U.S. shipyards as of 1981. The general trends and problems of CAD/CAM are well defined. Recent improvements in minicomputers work stations and software most likely make the specifics of this report obsolete.

FortenBerry, J.C.; Cox, J.F. "Multiple Criteria Approach to the Facilities Layout Problem." int. J. Prod. Res., vol.23, no.4 (JUL/AUG 1985) p. 773

Summary:

This paper describes an algorithm for facility layout. The prior algorithms for facility layout optimization used weighed flow and closeness values to determine best layout. Weights were determined throuqh empirical experience. This algorithm uses the A,E,I,O,U,X evaluation developed by Muther, flow, transit distance between activities, and the assumption that all entities are of equal size. The author claims that the assumption of equal size is reasonable in application. The use of a negative X value for entities that should be separated is more effective than the O or 1 value used by other algorithms.

Example:

The examples given are the proof examples from the other methods considered. In each case the "transit cost" was reduced by the authors algorithm. No full scale applications were described.

Application:

This approach provides a reasonable general alqorithm for shipyard facilities. The equal entity size assumption may make this algorithm useless for the optimum tool shed problem, but assignment of flow limits in addition to flow rates could chanqe the output sufficiently to give a good representation.

Since no discussion was provided in the paper of limitations on either the flow or transit distance values, it appears the algorithm can not include limits. This can be overcome by using simulation to model the algorithm and limitations.

Goemanov, L.A. “Industrial Production Simulation System Theoretical Approach” ACTA Polytech. Scand. no.Ma31 (1979), pq.66-70

Summary:

This paper presents a stochastic model for simulating industrial processes. The stochastic element of this model is created by Markov descriptions of network.

Example:

none given

Application:

This paper provides a theoretical model for process simulation under stochastic conditions. The mathematical representation is complete and other papers describing the application of this model to manufacturinq are cited. This paper provides proven approaches for consideration in stochastic modelinq.

Simulation of shipyard procedures is usually done on discrete, deterministic data. Detailed analysis of real world procedures will require some stochastic element to model the uncertainties involved in any major construction project. Specifically the use of stochastic models in just in time inventory management is necessary since the effect of random environmental influences can be a critical influence on just in time inventory management operating in the diffuse U.S. supply system (as compared to the geoqraphical nearness of management.) Other applications for stochastic modelinq include standard work packaqe optimization and initial construction facilities.

Graves, R. J.; Simth, J. M.; Kerbache,L. "Material Hand 1inq Problem Analysis Usinq Queueing Networks." Proceedings of the Annual Industrial Engineering Conference & Exposition, Louisville, KY (1983) P.269

Summary:
This paper describes a queueing network decomposition for planning a manufacturing facility. The algorithm used is discussed in Smith '85. A network with two subnets was used to describe the system. One subnet is the circulation network for non-instantaneous transitions of product. The other subnet is an activity, machines, and departments network. An open queueing model is used in the representation.

Example:
The example used in this paper describes the facility layout process used for converting an historic buildinq into a factory. The building shell was preserved as much as possible for aesthetic reason. This caused constraints on the distribution of activities and types of transportation systems, employed.

Application:
The facilities layout algorithm developed by Smith et.al. is demonstrated in a manufacturing environment. Althouqh this is an analytical algorithm vs. simulation model it does show flexibility and depth in this application and prior 1ibrary problem the model described could be used as the basis for a simulation model, as an alternative method for illustrative purposes, or as a macro level procedure in a planning hierarchy.

Karpowicz, A.S. and Simone, V. "A Contribution to Computer Simulation Methods, Application in Ship Production Process." Conference Proceedings of Ship Desiqn and Automation Meetinqs, North Holland Publication. pg.69-79

Summary:

This paper is a description of one attempt to write computer code for optimizing the scheduling of a product work breakdown system of ship construction. The simulation was based on two variables; material in tons, time in months. The model was deterministic.

The program was found to be useful but lackinq in detail.

Example:

The simulation of a ship's construction is used to demonstrate the proqram. The input was found to be to sparse for accurate simulation. Truncation of time units when lenqth of a process was not exactly a multiple of one month caused errors in time of process estimation. The material units did not allow qualitative distinctions.

AoDlication:

The program described in this paper does not have a sufficiently detailed field of variables to be used on the tool shed placement problem. The product unit, referred to as mar ine technology units (MTU's), is consistent with modern shipbuilding practices and proved usedful in the demonstration program. A more data rich MTU would have to be used on the tool shed problem.

Knapp, S.M. "A Floor Space Simulator for Shipyard Steel." Procedinqs SPC/IREAPS Technical Symposium, (23-25 AUG 1983) P. 501

Summary:

This paper is a description of a computer program that simulates the landinq of steel parts on a ship under construction. The program is a SPAR planning program which will interface with PERT-PAC scheduling proqrams without modificaiton. It is discrete and deterministic. Only minimal theory is presented.

Example:

This paper describes the approach taken in modelinq the activities to be simulated in the program. The proqram requires the programmer to define spaces, sub-spaces, and prohibited spaces in a relative 3 dimensional space(X,Y,Z), These spaces are filled with the parts and subassemblies defined by the programmer.

Application:

The program described has immediate application potential in the shipbuilding industry. Its output includes information on activity site, order, and duration for optimized production. When installed on a mainframe computer, the proqram can be used to plan new schedules in minutes after receivinq new information on the work completed.

Levy, F.K.; Thompson, G.L.; Wiest, J.S. “Multi-Shop, Workload Smoothinq Proqram”. Naval Research Loqistics Quarterly, VO1.9, no.1 (1963)

Summary:

This paper describes a schedule levelinq proqram. The critical path method is used with all events scheduled for regular early start (D). The program seeks the highest manhour demand day, selects events with slack manhour and delays these events. The result is a reduction in peak labor demand. This process is iterated until no improvement is made thru delays or no delays are possible for selected day.

Examples

The example, and proof, was based on fabricated data projected from historical values.

Application:

The general approach is similar to some facility layout algorithms in it’s sinqle comparison method of iterations. The nature of the calculations allows scale of smoothinq level to vary i.e. operational, tactical, strateqic. This can be achieved by creatinq a hierarchy with the nodes representinq work packaqes, then blocks, then completed products.

The use of this paper in the toolroom optimization proqram would be limited to the discussion of hierarchies since the proqram does not allow the depth of detail for eleqance of execution that more modern algorithms include.

Lewis; W.P. and Block, T.E. "On the Applications of Computer Aids to Plant Layout." International Journal of Production Research VO1.18, no.1 (JAN/FEB),pq.11-20

Summary:

This paper describes a procedure and computer program for optimizing facility layout. Various programs for facility layout are discussed. The two basic divisions of the programs are: 1) construction layout, and 2) optimizing layout.

Example:

No example is qiven, however the program descriptions include methodology for each proqrams facility layout alqorithm. The basic approach is to switch module sites then test to see if the system is more efficient. This approach can be very expensive computationally since it is exhaustive. Some discussion of computation costs is included.

Application:

The concepts for the algorithm described in this paper can be easily applied to shipyard use.

In the toolroom optimization problem the basic approach this alqorithm would take is; simulate a given feasible set of toolroom and work site locations to determine time spent on tool distributions next - toolroom relative size or site would be modified to some feasible value and the new layout would be simulated to determine time spent on tool distributions the layout with minimal distribution time would then be selected. This process would be exhaustively iterated within the feasible bounds and step sizes.

When usinq a limited set of feasible layouts in a job shop this approach can provide a rapid layout optimization.

Masri. S. and Moodie, C. “Usinq an Electronic Spreadsheet to Analyze Manufacturing Flow Systems.” Computers and Industrial Enqineering VO1.9, no.2 (1985)

Summary:

This paper describes the use of a microcomputer based spreadsheet to create small scale computer simulation. Desiqn of the model assumes the system described is a flow system. Job shop simulation can be accomplished by adding constraints to the model.

Example:

The paper uses a management resource planning model (MRP) for developing a bill of materials. The example proqram was oriented toward inventory management. A discussion of the relative merits of mainframe simulation VS . this spreadsheet based program is included in the paper.

Application:

This paper has no direct application in the resent study of shipyard use of simulation.

Future consideration miqht be given this low end approach to simulation for on site modification of process scheduling. The approach may provide sufficient computational ability for regular application of simulation to small shop operations.

McGinnis, L. and Graves, R. “A Decision Support System for the Outfit Planninq Problem.” Washington, D.C., Department of Transportation (1980,1982)

Summary:

This paper describes the work done to write a computer code for ship oufittinq optimization. The outfit problem is to maximize outfitting at the unit levels with on block outfitting considered next, and on board outfitting as least desirable, while operatinq within scheduling and facility constraints. Computer code, theory, and test data are included. Computer code is experimental, i.e. it is not optimized. A useful bibliography is included.

Example:

The example given in this paper is based on a computer generated model of a ship constructed in a product work breakdown structure environment (PWBS). The model qives a reasonable proof of the ability of the computer proqram to optimize an outfitting problem.

Application:

This paper offers a great deal of information on formulating proqrams of shipbuilding simulation. Six basic assumptions are made about the feasible range of values and the assumptions are shown empirically reasonable. Sufficient variables are described to provide detailed simlation.

The optimization is approached by runninq a selection proqram to find the optimum plan with no labor or facility scheduling constraints considered. The next step is to take the selected plan and check for violated scheduling constraints.

Okayama, Y. and Chirillo, L.S. “Product Work Breakdown Structure.” National Shipbuilding Research Proqram report 0164, Washington D.C., Department of Transportation (DEC 1982)

Summary:

This paper is a report on the management methodology used by IHI shipyard in Japan to classify and organize work so the the large construction project of shipbuilding can be divided into manageable work packages.

Example:

This report is an explanation of an operating management *system.* There are no illustrative examples given.

Replication:

The information in this report is the “state of the art” for shipyard management methods. It is a report on the methodology used to successfully employ group technology in shipbuilding.

The PWBS provides a format for organizinq work assignments. The basic breakdown is 3x2 matrix of work type. One side of the matriz is hull, outfitting, and painting work, the other side of the matrix is fabrication and assembly. These definitions form a 6 unit vector which is the third dimension added to the 4x4 product resources and product aspects matrix. Product resources are: material, manpower, facilities, and expenses. Product aspects are: systems area , zone, stage. By defininq all work packaqes with a 6x4x4 matrix, information about the work is rationalized and manageable.

The toolroom optimization project can use this work breakdown to define work packaqe tool needs. The PWBS is readily computerized and rich in detail.

Smith, J.M. and Bouanada, B. " Queueing Network Decomposition and Facilities Planning." Computer & Operations Research vol.12, no.1 (1985) pg. 1-16

Summary:

This paper describes a network based facilities planning optimization model. Two networks are formulated to created the model; one for product, one for facility. Two transaction levels are used; the aggregate between major nodes, detailed for within major nodes.

The algorithm is analytical and therefore faster than traditional simulation. However, it is also less detailed than simulation.

Example:

This paper uses the optimizing of a library facility as an example. The MNET V network lanquaqe was used along with Fortran V.

Application:

The algorithm in this paper is probably not directly. applicable to large scale problems encountered with the toolroom design problem. The network formulation may provide insight to the use of multiple networks to describe a single facility.

The paper was well documented, both on a theoretical and application experience level, it may provide leads to other approaches to the facility layout problem.

Appendix B

Other Applications

Shipbuilding

(1) Roper, Robert R. "The Shipbuilding Technology Transfer Proqram." Levinqston Shipbuilding Company, Oranqe, TX. File

Brief Description:

Levingston Shipbuilding Company has established a Technology Transfer Program (TTP) with Ishikawajima Harima Heavy Industries (IHI). The objective of the proqram is to (1) study IHI system, methods and techniques (2) compare practices and identify their applicability to American Shipyards (3) implement changes. The TTP is a third of the way throuqh the proqram and has been researching in the areas of Enqineerinq Desiqn, Planning and Production, Quality Insurances, and Industrial Enqineerinq. The utilization of computer-aid desiqn, standards, jiqs and fixtures, and accurate controls are a few of the significant findings of discrepancies.

(2) White, Stephen G. and Keyte, James H. "Computerized Shipboard Inventory Management: The STIMS Project". Marine Technology VO1.21, April 1984, pg. 170-5

Brief Description:

The Sun Transport Company is a marine transporting company. The company's Materials Management department beqan a process to automate its inventory functions. The computer system streamlines operations and also provides required management inforamtion to Maintenance, Operations, and other departments. The project focuses on shipboard inventory control, restocking and spare part control, The computer provides a process control system throuqh a data base, where each part is labeled with a numerical codinq system.

(3) Dr. Kuo, University of Strathclyde and Dr. MacCallum, university of Strathclyde. "Computer Aided Applications in Ship Technology". Computers in Industry. May 1984, pq. 211-219

Brief Description:

The article is a very good overview of shipyard knowledqe and computer applications. The computer applications are cross-referenced to specific journal in a very useful biblioqrapy.

(4) Bergstorm, Rabin P. "Computer-Aided SQC Makes Impact on Pontiac". Manufacturing Enqineerinq, May 1985, pq. 71+

Brief Description:
The Pontiac Motor Division has developed a computer-aided qaqinq information and statistical quality contro1 (SQC) proqram. The system required an interface of mechanical qaqinq operations and mini-computer production monitoring network at local levels. The system reduced, cost in scrap, warranty claims, operator inefficiency and eliminated lonq hand statistical quality control paperwork because the system spots the problem at its source.

(5) Mullins, Peter. "Perkins Aims at Fully Integrated Manufacture inq". Automotive Industry vol:164, January 1984, pq. 32

Brief Description:
Perkins produces diesel engine and has computerized *most* facets of the company. Computer systems currently beinq used include: Computer-Aided Design and Manufacturing, Data Base system linked to IBM, and ASPRO (Assembly Proqramminq). The system was most rewardinq within the central computer system such that information can be translated quickly for continous monitoring of materials.

(6) "Advanced Manufactuinq Technology". Toolinq and Production V01:50, September 1985, pg.96

Brief Description:
Chrysler assembly plant is applyinq control systems by using micro-processor based resistance welding controls manufactured by Medar Inc. Farminqton Hills, MI. The expected benefits are reduced downtime by helpinq maintenance personnel quickly pinpoint problems.

(7) Society of Automotive Enqineerinq. "Computer-Aided Enqineering :a Step Beyond". Automotive Enqineering . V0l:92 March 1984, pq. 30-3

Brief Description:
A general article that describes how computers can assist design efforts and manufactuinq. However, the article's main emphasis is on technology selection, which is rather useful.

Aerospace

(8) Whitinq, Frank T. "Computer Systems". Astronauts and Aerospace V01:21, December 1983, pg. 59+

Brief Description:
The article describes a number of computer systems with special interest in the computer-aided design systems. Upon adaption to the system, the biggest savinqs appear to come from the design revisions and data base. The data base allows quick retrieval of information and feedback on the design.

(9) "From Design to Flight, Computers Guide Aerospace Industry". Aviation Week and Space Technology V01:120 Auqust 13,1984, pg. 89-148

Brief Description:
Computer systems are integrated into controls, desiqn, and inspection. The article discussed the CAD systems as beinq useful tools in simulating aircraft superstructure stress. In addition, the aerospace industry *is* increasing their use in mini-computers and micro-computers because of their growinq capabilities.

Manufacturing

(l0) Advanced Manufacturing Technology. "Computer Proqramminq Pays Off". Tooling and Production V01:121 September 1984

Brief Description:
Precise Tool and Mold Company, a manufacturing company for the plastics industry, has used computers to assist in mold-makinq and developing NC tapes quickly and efficiently. The article describes how they installed a computer qraphic system into a fairly small company.

(11) Quinian, Joseph C. "Flexible Assembly-manufacturinq's newest frontier" Tool and Production V01:121 Septemer 1984 pq. 33

Brief Description:
The article defines three forms of robotics: cell, flexible and FDS. It dicuss the benefits and cost of robotics and how a company may approach integrating such equipment.

User’s Manual

2.1 Introduction

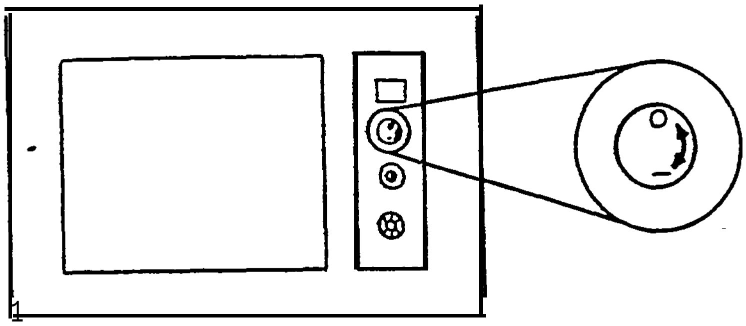
The following is the CATD User’s Manual for the IBM-AT Computer and the IBM-PC with 8087 Co-Processor Chip. It will provide the user with a step-by-step guide for operating the computer and running the CATD proqram.

2.2 Operating the Computer

2.2.1 How To Get Started

(1) Turn on the power to the system unit by flippinq on the red rocker switch which is located on the power strip. The power strip is the plug-in device for the computer. The strip *is* usually attached to the underneath side of the desk, or behind the computer. If you have problems findinq the strip, just follow the cord of the computer to the strip. The rocker switch will be lit when the system is turned on.

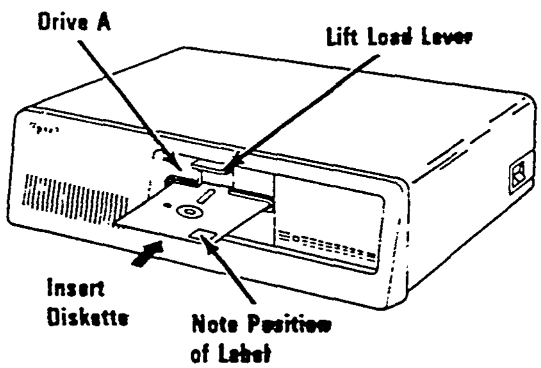
(2) If the unit is already on, but the display or screen is blank, turn the top knob to the right of the screen clockwise.



knob

Screen

(3) Insert the disk containing the CATD executable into the disk drive. Diskettes are loaded into the drive with the label-side of the disk facing the liqht of the disk drive. Turn the lever over the disk slot to secure the disk in the drive.



Disk Drive

(4) The computer screen will display A:\> or C:\>, which

indicates that the computer is ready for you to type somethinq in. You want the computer to display A:\>. Hence, if C:\> is displayed, simply type A:.

(5) Once you have received an A:\> on the screen, type DIR, which is the disk directory. You should see:

CATD EXE 97608 7-28-86 7:28P

If you do not see CATD EXE listed in the directory, you either have the wrong diskette or the CATD Program is listed under another name. In any cases, consult your technician concerning the diskette.

(6) If you wish to get a print-out of the program you must activate the printer before You actually run the program. See the next section on How to Get a Print-Out.

(7) To Start the program without a print-out just type CATD and hit return.

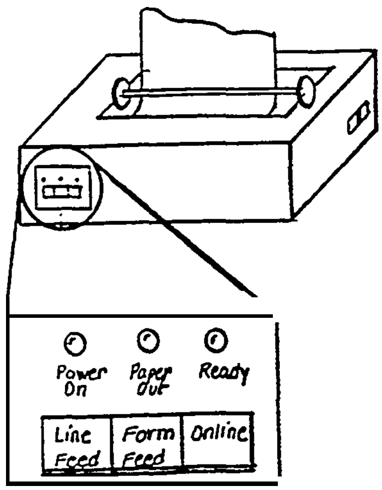
2.2.2 How To Get A Print-Out

(1) Turn on the power to the printer by flippinq the on-switch which is located on the riqht-hand side of the printer.

(2) The power panel is located on the front of the printer, in which there are three controls and three liqhts. The lights indicate the following: Power On, Paper Out, Ready.

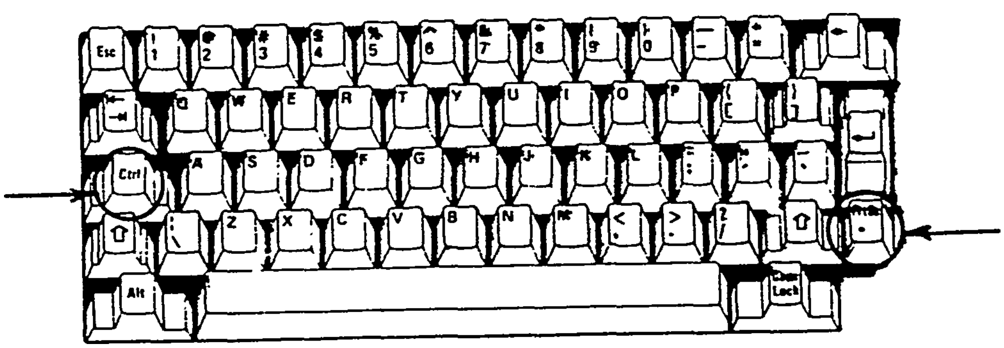
The power on light and the ready light must be lit to start the printer. If the ready light is not on, press the thin key directly below the ready light to reset the printer.

Printer



Controls

(3) Type CATD, but before hittinq the return, press the Cntrl key and the Prt Sc key simultaneously, and then hit return. At this. Point the printer will print all the characters on the screen until You inactivate the printer. A display of the keyboard is shown on the next paqe.



Keyboard

(4) You should see the beginning of the program and hear the printer running. If the printer is not operating, exit from the Program and try typing CATD Cntrl PrtSc again. If you have difficulties ask your technician for help. It is possible that the printer is not workinq correctly.

(5) After you are throuqh usinq the proqram, press Cntrl and PrtSc keys to stop the printer.

(6) To get the paper out of the printer, press the paper cut button on the power panel of the printer.

2.3 Running the Program

2.3.1 Entering Data

Once YOU have the CATD proqram started, the program will prompt You for all the needed input. The followinq section shows you how to enter Your data into the program.

The program will ask You to choose one of the following options:

1- TOOLROOM STATISTICS

2- TOOLROOM ECONOMICS

3- OPTIMAL LOCATION

The first option gives statistical serving or queueing information for the toolroom. The second option finds the cost of Serving each toolroom. Lastly, the third option determines a best location for the toolroom.

The screen will next display:

INTEGER

ENTER :

The ENTER prompts you for an input. The INTEGER indicates that the number you type in must be a whole number, therefore exclude the decimal point. For example, if you were to choose option two and typed in 2.0 instead of 2, the screen would display:

ERRORMISTYPED DATA

1- TOOLROOM STATISTICS

2- TOOLROOM ECONOMICS

3- OPTIMAL LOCATION

INTEGER

ENTER :

The program qives you your options again and prompts you for an input. Hence, YOU must type l,2,or 3.

After each input, the computer displays:

ENTER 1 TO CORRECT ELSE <RETURN> TO CONTINUE

This statement allows YOU to change the data YOU just entered. Suppose you choose option TWO, and therefore typed in 2. However, you realized that YOU made a mistake and wanted option one. YOU would simply type a 1 after this command and the program would list your options again. If you wish not to make any chanqes just hit return and go on to the next input statement.

After choosing an option, the program will ask YOU:

INPUT THE NUMBER OF TOOLROOMS

INTEGER

ENTER :

Type in the number of toolrooms you want to analyze. The program can handle ten different toolrooms, So You should enter a whole number between 1 and 10.

The program will ask you questions Concerninq each toolroom:

INPUT THE NAME OF TOOLROOM #1:

CHARACTER

ENTER S

The CHARACTER ENTER tells you that the computer wants alphabetic input. The computer can hold 20 letters. Hence, if you type more than 20 letters, the name will be trunaicated after 20 letters. If the first toolroom is called track-3, type TRACK-3. The Computer will ask YOU questions about the first toolroom or track-3:

INPUT THE FOLLOWING DATA FOR TRACK-3

NUMBER OF WINDOWS

INTEGER

ENTER :

YOU WOUld type in the number 0f windows in the remembering tht integer indicates a whole number input. you will next be asked to input:

NUMBER OF ATTENDANTS

INTEGER

ENTER :

Again, just enter the appropriate data, and hit return. Next , the computer will display:

MEAN SERVICE

REAL

ENTER :

This input request is different than the previous request in the respect that the program is asking for a real number input. This means that a decimal number is expected. For example, if the mean service rated is 1.48, YOU must type 1.48. If you were to type 148, the program would not give You an error messaqe. However? you would be entering in misleading data, since 1.48 is not equal to 148.

Finally, YOU will be asked to enter the mean arrival rate. Just enter the your data, keeping in mind what type of data is to be inputted: integer, real, or character. This is the minimum amount of input required to run the first option. If you had chosen option one, the program would print out the statistical data for each toollroom. After the print out of each tooroom the computer will ask:

CHOOSE THE FORM TO DISPLAY P(N)

1- TABLE

2- CHART

The first option will give you a listing of the probability of the number of units in the service area. The second option prints out a bar graph of the probability.

After all the statistical data is printed out the computer will display:

ENTER ONE OF THE FOLLOWING TO:

1- CONTINUE WITH THE MODEL

2- RETURN TO STARTUP MENU

3- QUIT

The first option will give YOU the statistical data again with your option of display. The second option will return YOU the to beqinning of the program. The third option will exit you from the proqram. If You choose to start the program again to analyze the economics or optimal location of the toolroom, the program will save the data You have already entered. The Computer will display:

THE NUMBER OF TOOLROOMS IS 2.

ENTER 1 To CORRECT ELSE <RETURN> TO CONTINUE

If the data is incorrect or you wish to change the data? type a 1. The proqram will ask YOU to enter the correct input:

INPUT THE NUMBER OF TOOLROOMS

INTEGER

ENTER :

The proqram will ask you various questions about each toolroom dependinq **on what options, you** choose. The following is a list of data requirements for the different options:

All 0ptions

* Number of Toolrooms
* Name of Toolrooms

For Each Toolroom:

* Number of Attendants
* Mean Service Rate
* Mean Arrival Rate

Option Two

* Number of Service Areas
* Names of the Service Areas
* Distances between Service Area and Toolroom
* Buildinq Cost

* Inventory Cost
* Wage of Workers
* Number of Workers

Option Three

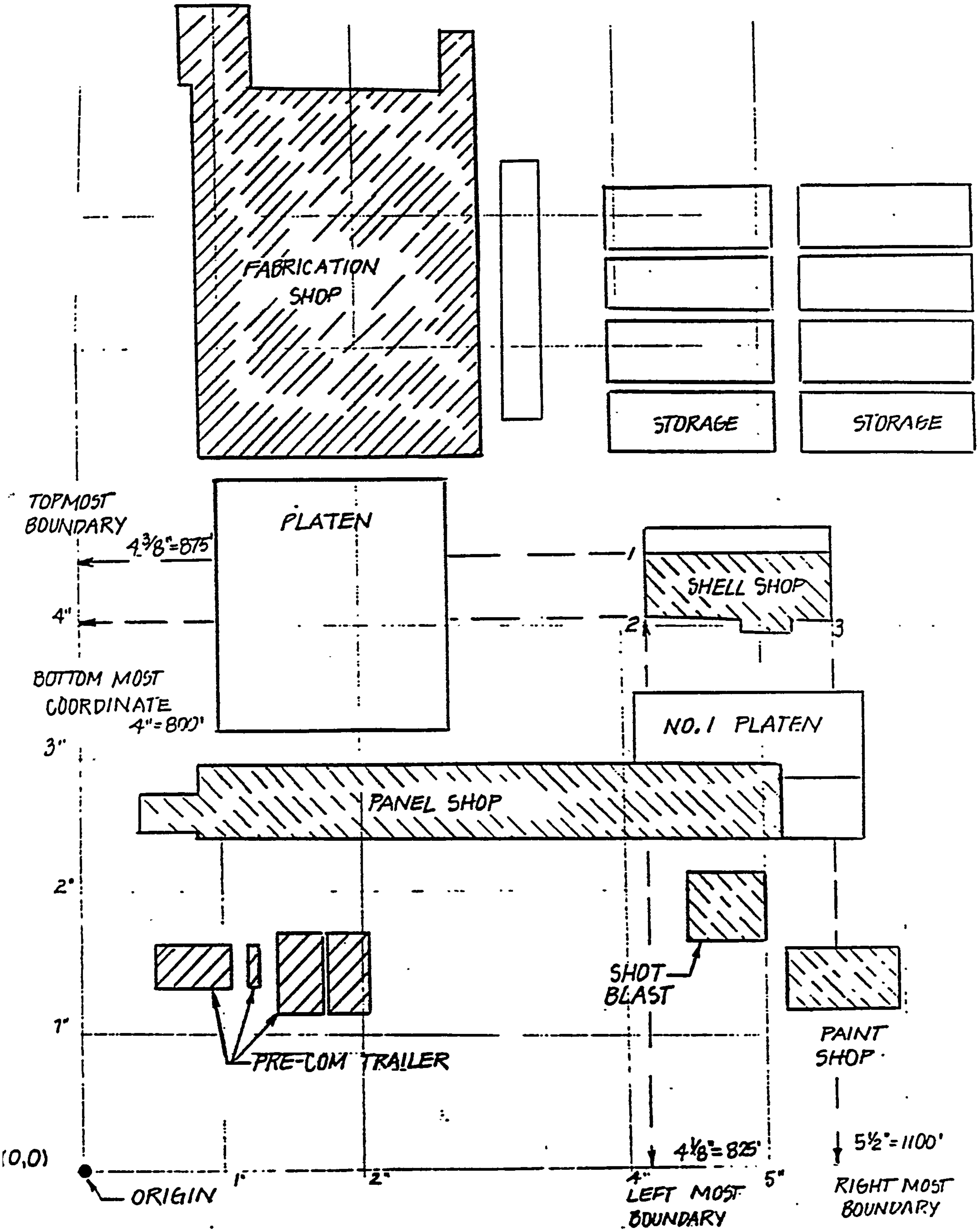
* Number of Service Areas
* Names of Service Areas
* Coordinates of the Service Areas

You should have all of this data prior to using the CATD proqram. However$ based on the results of the analysis YOU may be asked to gather additional coordinates of the infeasible areas.

2.3.2 Obtaining Coordinates

In order to, determine the coordinates of the areas, you need a drawing or layout of the shipyard that has all the buildings specified. You must locate all the service areas and infeasible areas on the layout. Once YOU have found the areas, choose an orgin or a common startinq point from which you will measure all the distances of the service areas. The origin is best placed in the lower lefthand side of the service areas to avoid obtaininq negative coordinates (See Fiqure 1) .

In Fiqure 1, the orgin is located on the bottom of the layout and is labeled zero. Lightly pencil in the two lines! where one line runs horizontally and the other vertically. If grid lines are not already drawn on your layout, it may be



FABRICATION SHOP
STORAGE
STORAGE
TOPMOST BOUNDARY
4 3/8" = 875'
PLATEN
1
SHELL SHOP
4"
2
3
BOTTOM MOST COORDINATE
4" = 800'
NO. 1 PLATEN
3"
PANEL SHOP
2"
SHOT BLAST
1"
PRE-COM TRAILER
PAINT SHOP
4 1/8" = 825'
5 1/2" = 1100'
(0,0)
ORIGIN
1"
2"
4"
5"
LEFT MOST BOUNDARY
RIGHT MOST BOUNDARY

FIGURE 1

advisable to pencil the grid lines in for scaling purposes. In this examples the grid lines are 1 inch apart representing 200 feet in distance. The scale to right indicates that 1″=200 which establishes the dimensions of grid lines. You should also establish the size of your grid lines with the scale on Your layout. In figure 1, the coordinates of the service area was found by labeling the corners of the service area and measuring the distances. To find the x-coordinates, take a ruler and measure the distances from the horizontal line to corner 1 and corner 3 and record the inches. Next, convert the inches into feet with respect to the specified scale. Hence, the x-coordinate of the right most boundary is 1100 feet, and the x-coordinate of the left most boundary is 825 feet. To find the y-coordinate measure the distances between the horizontal line and corners 1 and corner 2. The y-coordinate of the top most boundary is 875 feet, and the bottom most boundary is 800 feet. You will need to find the coordinates for each service area and infeasible area.

2.3.3 Tutorial

The following section is a tutorial from an actual session. It may be helpful to use this tutorial as a visual aid during your first sessions with the CATD.

```
******************************************************************************

                                   ***CATD***

******************************************************************************



PLEASE ENTER THE APPROPRIATE INTEGER
 TO CHOOSE ONE OF THE FOLLOWING OPTIONS
         1 - TOOLROOM STATISTICS
         2 - TOOLROOM ECONOMICS
         3 - OPTIMAL LOCATIONS
INTEGER
ENTER:
1



INPUT NUMBER OF TOOLROOMS
INTEGER
ENTER:
1

ENTER 1 TO CORRECT ELSE <RETURN> TO CONTINUE



INPUT NAME OF TOOLROOM #  1:
CHARACTER
ENTER:
FABROC SHOP

ENTER 1 TO CORRECT ELSE <RETURN> TO CONTINUE



INPUT THE FOLLOWING DATA FOR FABRIC SHOP


    NUMBER OF WINDOWS
INTEGER
ENTER:
2

ENTER 1 TO CORRECT ELSE <RETURN> TO CONTINUE



    NUMBER OF ATTENDANTS
INTEGER
ENTER:
1

ENTER 1 TO CORRECT ELSE <RETURN> TO CONTINUE
```

```
    MEAN SERVICE RATE
REAL
ENTER :
1.25

ENTER 1 TO CORRECT ELSE! < RETURN> TO CONTINUE



    MEAN ARRIVAL RATE
REAL
ENTER :
201

ENTER 1 TO CORRECT ELSE <RETURN> TO CONTINUE
 .


***STATISTICS FOR FABRIC SHOP


****************************************************************************
    MEAN SERVICE RATE -       1.25
    MEAN ARRIVAL RATE -        .20
    MEAN TIME SPENT WAITING IN LINE -      3.84
    MEAN TIME SPENT WAITING IN SYSTEM -      4.64
    PROBABILITY OF EMPTY QUEUE -      5.22
    UTILIZATION FACTOR -        16
    EXPECTED NUMBER OF UNITS IN SYSTEM -       .93
    MEAN LENGTH OF QUEUE -       .77
****************************************************************************


CHOOSE THE FORM TO DISPLAY P(N)
        1 - TABLE
        2 - CHART
1


NUMBER OF UNITS  PROBABILITY  OF  OCCURRENCE

        1                 8392
        2                  1349
        3                 .01217
        4                 .0035
        5                 .0006


ENTER ONE OF THE FOLLOWING TO:
        1 - CONTINUE WITH MODEL
        2 - RETURN TO STARTUP MENU
        3 - QUIT
INTEGER
ENTER :

2
PLEASE ENTER THE APPROPRIATE INTEGER
 TO CHOOSE ONE OF THE FOLLOWING OPTIONS
        1 - TOOLROOM STATISTICS
        2 - TOOLROOM ECONOMICS
        3 - OPTIMAL LOCATIONS
INTEGER
ENTER:
2
```

```
CURRENT NUMBER OF TOOLROOMS IS 1

ENTER A 1 TO CHANGE ELSE <RETURN> TO CONTINUE


TOOLROOM NAME IS FABRIC SHOP

ENTER 1 TO CHANGE ELSE <RETURN>. TO CONTINUE


INPUT THE FOLLOWING DATA FOR FABRIC SHOP

    NUMBER OF WINDOWS IS 2

ENTER 1 TO CHANGE ELSE <RETURN> TO CONTINUE


    NUMBER OF ATTENDANTS IS 1

ENTER 1 TO CHANGE ELSE <RETURN> TO CONTINUE


    MEAN SERVICE RATE IS 1.2500

ENTER 1 TO CHANGE ELSE <RETURN> TO CONTINUE


    MEAN ARRIVAL RATE IS    .2010

ENTER 1 TO CHANGE ELSE <RETURN> TO CONTINUE


    BUILDING COST
REAL
ENTER :
44000.

ENTER 1 TO CORRECT ELSE <RETURN> TO CONTINUE


    INVENTORY COST
REAL
ENTER :
99000.

ENTER 1 TO CORRECT ELSE <RETURN> TO CONTINUE


    WORKERS AVERAGE HOURLY WAGE
REAL
ENTER:
15.0
```

```
ENTER 1 TO CORRECT ELSE <RETURN> TO CONTINUE



    AVERAGE WALK RATE OF WORKERS
REAL
ENTER :
210.

ENTER 1 TO CORRECT ELSE <RETURN> TO CONTINUE



INPUT NUMBER OF SERVICE AREAS
INTEGER
ENTER :
3

ENTER 1 TO CORRECT ELSE <RETURN> TO CONTINUE



INPUT NAME OF SERVICE AREA # 1:
CHARACTER
ENTER :
FAB SHOP

ENTER 1 TO CORRECT ELSE <RETURN> TO CONTINUE



INPUT NAME OF SERVICE AREA # 2:
CHARACTER
ENTER:
SHELL SHOP

ENTER 1 TO CORRECT ELSE <RETURN> TO CONTINUE



INPUT NAME OF SERVICE AREA # 3:
CHARACTER
ENTER :
PANEL SHOP

ENTER 1 TO CORRECT ELSE <RETURN> TO CONTINUE



INPUT THE AVERAGE NUMBER OF WORKERS THAT USE FABRIC SHOP
REAL
ENTER :
97.

ENTER A 1 TO CORRECT ELSE <RETURN> TO CONTINUE



ENTER THE AVERAGE DISTANCE BETWEEN:
TOOLROOM: FABRIC SHOP
SERVICE AREA:FAB SHOP
REAL
ENTER :
800.
```

```
ENTER 1 TO CORRECT ELSE <RETURN> TO CONTINUE

TOOLROOM: FABRIC SHOP
SERVICE AREA:SHELL SHOP
REAL
ENTER :
400.

ENTER 1 TO CORRECT ELSE <RETURN> TO CONTINUE

TOOLROOM: FABRIC SHOP
SERVICE AREA:PANEL SHOP
REAL
ENTER:
450.

ENTER 1 TO CORRECT ELSE <RETURN> TO CONTINUE



***STATISTICS FOR FABRIC SHOP


**********************************************************************************
    MEAN SERVICE RATE -     1.25
    MEAN ARRIVAL RATE -      .20
    MEAN TIME SPENT WAITING IN LINE -      3.84
    MEAN TIME SPENT WAITING IN SYSTEM -      4.64
    PROBABILITY OF EMPTY QUEUE -     5.22
    UTILIZATION FACTOR -      .16
    EXPECTED NUMBER OF UNITS IN SYSTEM -       .93
    MEAN LENGTH OF QUEUE -      .77
**********************************************************************************


CHOOSE THE FORM TO DISPLAY P(N)
         1 - TABLE
         2 - CHART
2


FLOT OF P(N) - PROBABILITY OF N UNITS IN SYSTEM



               100





                  ************
                80************
PROBABILITY       ************
 (PERCENT)        ************
                  ************
                  ************
                  ************
                60************
                  ************
                  ************
                  ************
                  ************
                  ************
                40************       2.17
                  ************
                  ************
```

```
            ************
            ************
            ************
          20************
            ************
            ************************
            ************************
            ************************
            ************************
                 1           2           3           4           5

                                UNITS


******************************************************************************
OPTIMAL MEAN SERVICE RATE    -       1.1839
WAITING COST PER UNIT         -          57.67
SERVICING COST PER UNIT       -          12.00
TOOLROOM COST DUE TO QUEUEING-       20655.58
TOOLROOM COST TOTAL          -      163655.60
******************************************************************************


COST  FOR  ALL  TOOLROOMS  -      163655.60
******************************************************************************




ENTER ONE OF THE FOLLOWING TO:
         1 - CONTINUE WITH MODEL
         2 - RETURN TO STARTUP MENU
         3 - QUIT
INTEGER
ENTER :
2
PLEASE ENTER THE APPROPRIATE INTEGER
 TO CHOOSE ONE OF THE FOLLOWING OPTIONS
         1 -  TOOLROOM STATISTICS
         2 - TOOLROOM ECONOMICS
         3 -  OPTIMAL LOCATIONS
INTEGER
ENTER :
3



CURRENT NUMBER OF TOOLROOMS IS 1

ENTER A 1 TO CHANGE ELSE <RETURN> TO CONTINUE


TOOLROOM NAME IS FABRIC SHOP

ENTER 1 TO CHANGE ELSE <RETURN> TO CONTINUE


INPUT THE FOLLOWING DATA FOR FABRIC SHOP


    NUMBER OF WINDOWS IS 2
```

```
ENTER 1 TO CHANGE ELSE <RETURN> TO CONTINUE


    NUMBER OF ATTENDANTS IS 1 .

ENTER 1 TO CHANGE ELSE <RETURN> TO CONTINUE


    MEAN SERVICE RATE IS  1.2500

ENTER 1 TO CHANGE ELSE <RETURN> TO CONTINUE


    MEAN ARRIVAL RATE IS    .2010

ENTER 1 TO CHANGE ELSE <RETURN> TO CONTINUE


    BUILDING COST IS       44000.00

ENTER 1 TO CHANGE ELSE <RETURN> TO CONTINUE


    INVENTORY COST IS       99000.00

ENTER 1 TO CHANGE ELSE <RETURN> TO CONTINUE


INPUT NUMBER OF WORKERS
INTEGER
ENTER:
97

ENTER 1 TO CORRECT ELSE <RETURN> TO CONTINUE


    WORKERS AVERAGE HOURLY WAGE IS   15.00

ENTER 1 TO CHANGE ELSE <RETURN> TO CONTINUE


    AVERAGE WALK RATE OF WORKERS IS 210.00

ENTER 1 TO CHANGE ELSE <RETURN> TO CONTINUE


NUMBER OF SERVICE AREAS IS 3

ENTER 1 TO CHANGE ELSE <RETURN> TO CONTINUE


INPUT NAME OF SERVICE AREA #  1:
CHARACTER
ENTER:
EAR SHOP
```

```
ENTER 1 TO CORRECT ELSE <RETURN> TO CONTINUE


INPUT THE FOLLOWING DATA FOR SERVICE AREA: FAB SHOP


    X-COORD. OF LEFTMOST BOUNDARY
REAL
ENTER :
150.


    Y-COORD. OF BOTTOMMOST B0UNDARY
REAL
ENTER :
1050.


    X-COORD. OF RIGHTMOST BOUNDARY
REAL
ENTER :
600.


    Y-COORD. OF TOPMOST BOUNDARY
REAL
ENTER :
1700.

ENTER 1 TO CORRECT ELSE <RETURN> TO CONTINUE


INPUT NAME OF SERVICE AREA # 2:
CHARACTER
ENTER :
SHELL SHOP

ENTER 1 TO CORRECT ELSE <RETURN> TO CONTINUE


INPUT THE FOLLOWING DATA FOR SERVICE AREA: SHELL SHOP


    X-COORD. OF LEFTMOST BOUNDARY
REAL
ENTER :
825.


    Y-COORD. OF BOTTOMMOST BOUNDARY
REAL
ENTER :
800 .


    X-COORD. OF RIGHTMOST BOUNDARY
REAL
ENTER :
1000.
```



```
    Y-COORD. OF TOPMOST BOUNDARY
```

```
    Y-COORD. OF TOPMOST BOUNDARY
REAL
ENTER:
925.

ENTER 1 TO CORRECT ELSE <RETURN> TO CONTINUE



 INPUT NAME OF SERVICE AREA # 3:
CHARACTER
ENTER :
PANEL SHOP

ENTER 1 TO CORRECT ELSE <RETURN> To CONTINUE



 INPUT THE FOLLOWING DATA FOR SERVICE AREA: PANEL SHOP


    X-COORD. OF LEFTMOST BOUNDARY
REAL
ENTER :
200.


    Y-COORD. OF BOTTOMMOST BOUNDARY
REAL
ENTER:
500.


    X-COORD. OF RIGHTMOST BOUNDARY
REAL
ENTER :
1025.


    Y-COORD. OF TOPMOST BOUNDARY
REAL
ENTER:
600.

ENTER 1 TO CORRECT ELSE <RETURN> TO CONTINUE



THE AVERAGE NUMBER OF WORKERS THAT USE FABRIC SHOP
IS       97.

ENTER A 1 TO CHANGE ELSE <RETURN> TO CONTINUE



INPUT THE NUMBER OF WORKERS IN SERVICE AREA FAB SHOP
INTEGER
ENTER:
22

ENTER 1 TO CORRECT ELSE <RETURN> TO CONTINUE



INPUT THE NUMBER OF WORKERS IN SERVICE AREA SHELL SHOP
INTEGER
ENTER:
```

```
15

ENTER 1 TO CORRECT ELSE <RETURN> TO CONTINUE



INPUT THE NUMBER OF WORKERS IN SERVICE AREA PANEL SHOP
INTEGER
ENTER :
13

ENTER 1 TO CORRECT ELSE <RETURN> TO CONTINUE



INPUT THE NUMBER OF INFEASIBLE AREAS
(INTEGER)
ENTER :
1


NUMBER OF INFEASIBLE AREAS IS:    1

ENTER 1 TO CHANGE ELSE <RETURN> TO CONTINUE



INPUT THE FOLLOWING DATA FOR  INFEASIBLE AREA # 1
    X-COORD. OF LEFTMOST BOUNDARY
REAL
ENTER:
200.
    Y-COORD. OF BOTTOMMOST BOUNDARY
REAL
ENTER:
550.
    X-COORD. OF RIGHTMOST BOUNDARY
REAL
ENTER:
550.
    Y-COORD. OF TOPMOST BOUNDARY
REAL
ENTER:
950.
INFEASIBLE AREA # IS BOUNDARIES ARE:
    X-COORD. OF LEFTMOST BOUNDARY IS 200.0000
    Y-COORD. OF BOTTOMMOST BOUNDARY IS 550.0000
    X-COORD. OF RIGHTMOST BOUNDARY IS 550.0000
    Y-COORD. OF TOPMOST BOUNDARY IS 950.0000

ENTER 1 TO CHANGE ELSE <RETURN> TO CONTINUE



***STATISTICS FOR FABRIC SHOP


***OPTIMAL LOCATION IS (438.7500,950.0000)


********************************************************************************
    MEAN SERVICE RATE -      1.25
    MEAN ARRIVAL RATE -       .20
    MEAN TIME SPENT WAITING IN LINE -      3.84                       2.22
    MEAN TIME SPENT WAITING IN SYSTEM -      4.64
    PROBABILITY OF EMPTY QUEUE -      5.22
```

```
    UTILIZATION FACTOR -       .16
    EXPECTED NUMBER OF UNITS IN SYSTEM -       .93
    MEAN LENGTH OF QUEUE -       .77
********************************************************************************


CHOOSE THE FORM TO DISPLAY P(N)
         1 - TABLE
         2 - CHART
1


NUMBER OF UNITS   PROBABILITY OF OCCURRENCE

        1                    .8392
        2                    .1349
        3                    .0217
        4                    .0035
        5                    .0006


********************************************************************************
OPTIMAL MEAN SERVICE RATE   -       1.1839
WAITING COST PER UNIT        -           57.67
SERVICING COST PER UNIT      -           12.00
TOOLROOM COST DUE TO QUEUEING-        20585.74
TOOLROOM COST TOTAL         -       163585.80
********************************************************************************


COST FOR ALL TOOLROOMS      -      163585.80
********************************************************************************


ALL TOOLROOM LOCATIONS ARE OPTIMAL


ENTER ONE OF THE FOLLOWING TO:
         1 - CONTINUE WITH MODEL
         2 - RETURN TO STARTUP MENU
         3 - QUIT
INTEGER
ENTER :

3
Pause.
Please press <return> to continue.
```

2.4 Interpreting the Results

Once YOU have run the program it is important to interpret the output data. If You are analyzing the optimal location of a present toolroom, the user should compare the cost of queuing for the optimal toolroom and the present toolroom to estimate the possible savings. If you are evaluating the feasibility of an additional toolroom, you should note the Cost of queuinq for each alternative.

Technical Manual

3.1 Introduction

This is the CATD Technical Manual. Its purpose is to provide the details necessary to maintain and upgrade the CATD program. The intended audience for this manual is a person with a few years of software experience knows FORTRAN77 and familiar with running and developing software on microcomputers.

The CATD tool is to be used as an aid for choosing optimal locations of toolrooms in a shipyard. Its limitations should be understood since the results of the program can be no better than the inputted data.

3.2 Design Process

The design of the program followed structured analysis and structured design methodologies as described in Structured **Analysis and System Specification by Tom de Marco (1) and The Practical Guide to Structured** Systems **Design** by Meilir Page-**Jones (2).**

CATD program was designed using structured design methods. These methods improve the quality of the software and reduce the numbers of errors that can surface durinq test and debug periods and later during maintenance. They improve the probability that the software will perform as required.

The writing of the software consisted of three phases: requirements design, and code. Each of these phases use different aspects of structured methods and are described in the following sections.

3.2.1 Requirements

The use of structured analysis methods assist in defining the problem and its solution. The designer must first consider the overall function of the system. In this case, the function of the system would be choosing optimal toolroom locations. After the function of the system has been determined, the system can be broken into sub-systems. These sub-systems should put similiar items into blocks. For this reason, the system has been broken into five blocks or modules: (1) Get Valid Input (2) Set Up Shipyard Model (3) Do Queuing Theory (4) Calculate Cost of Shipyard Model (5) Output Shipyard Model, and Statistics. Each of these modules can be broken down further and further until a simple function *is* processed.

To limit the complexity of each module, the modules are refined into even sma1ler modules that serve only one function. Upon establishing what modules are required to serve each sub-system, data flow diagrams are used to show the flow of data through the system. In Appendix A, you will find data flow diagrams for the entry system and for each sub-system. In addition you will see a data dictionary which clarifies the abbreviated names of the data. Decomposition of the data flow diagram levels the design allowinq the designer to portray the system at various levels of detail. This decomposition prevents overly busy diagrams and aids in quickly grasping what is going on in the system.

3.2.2 Design

The use of structured design methods assist in transforming the data flow diagrams that represent the concept of a system into the code. During this phase, data flow diagrams are transformed into structured charts. The structured charts are enclosed in Appendix B. From the flow diagrams, the designer is able to see the data dependency of each sub-system and its modules. Hence, control characteristic can be uncovered and help define interfaces between the modules.

3.2.3 Code

Once the structured charts are drawn, a psuedo-language format is first used to develop code. The designer describes each module depicted in the structured charts using the psuedo-code. This phase is found in Appendix C, and is referred to as process specification. The process specification outlines the algorithm of the system, where the order of the modules are determined by the structured charts.

After the algorithm is developed the psuedo-code is translated into the target language in this cases, Fortran77. Each part of the algorithm must be translated into a group of Fortran statements. The final code is included in Appendix D, where each module is documented for clarity.

3.3 Correcting a Discovered Error

While structured methodolgies decrease the chance of bugs occurring, they cannot completely prevent their occurrence. When an error is discovered, the technician has various ways to approach the problem depending on the type of bug and the tools available for debugginq. If a symbolic debugger is available, the error can be traced relatively easily. Without such a too l, adding write statements to output variables at different steps in the program can be used. When the reason for the error is discovered, the technician should correct the error and document its correction. It may be necessary to change the structured charts, the data flow diagrams, and/or the process specifications depending on the severity of the error.

3.4 How to Improve or Change a Feature

When it becomes necessary to tailor the program for particular needs or enhance the program to increase its applicability, one must begin with the data flow diagrams. It is strongly suggested that the person responsible for modifying the code follow these steps: (1) Identify the domain of change on the data flow diagrams. (2) Define the suggested change using the data flow diagrams and data dictionary. (3) Insert the change into the previous data flow diagrams, and note any changes to the leveling of the system. (4) Carry this design change through the structured charts. (5) Beware of dependent data modules and interface the modules accordingly. (6) Carry this design change through the process specifications and finally, Fortran77. It is advisable that the actual code be the last thing changed in this situation. Tampering with an involved system carelessly could lead to errors that are difficult to detect.

3.5 Data

Prior to using the CATD program, toolroom data needs to be gathered. The data can be broken into four categories: financial, travel, worker and queueing information. Care must be taken when gathering data to avoid misleading results of the program.

3.5.1 Financial Information

The financial data primarily consist of building and inventory cost of each toolroom. Also, any possible expansions to the toolroom should be documented. In addition, the average salary of the workers should be determined for each toolroom, including fringe benefits.

3.5.2 Travel Information

The travel data is based on the distances between the toolroom and each service area. You should keep all distance measurements consistent, in which rounding to the nearest tens of feet or meters is acceptable. Include the average walking time of a normal worker, where allowances for fatigue, personal and delay factors should be considered. The walking rate will be in units of distance/time or feet/second.

3.5.3 Worker Information

The average number of workers or customers who are serviced in the toolroom during one day should be recorded. Depending on the traffic in the toolroom, the queuing study and the number of customers being served could be tabulated at the same time.

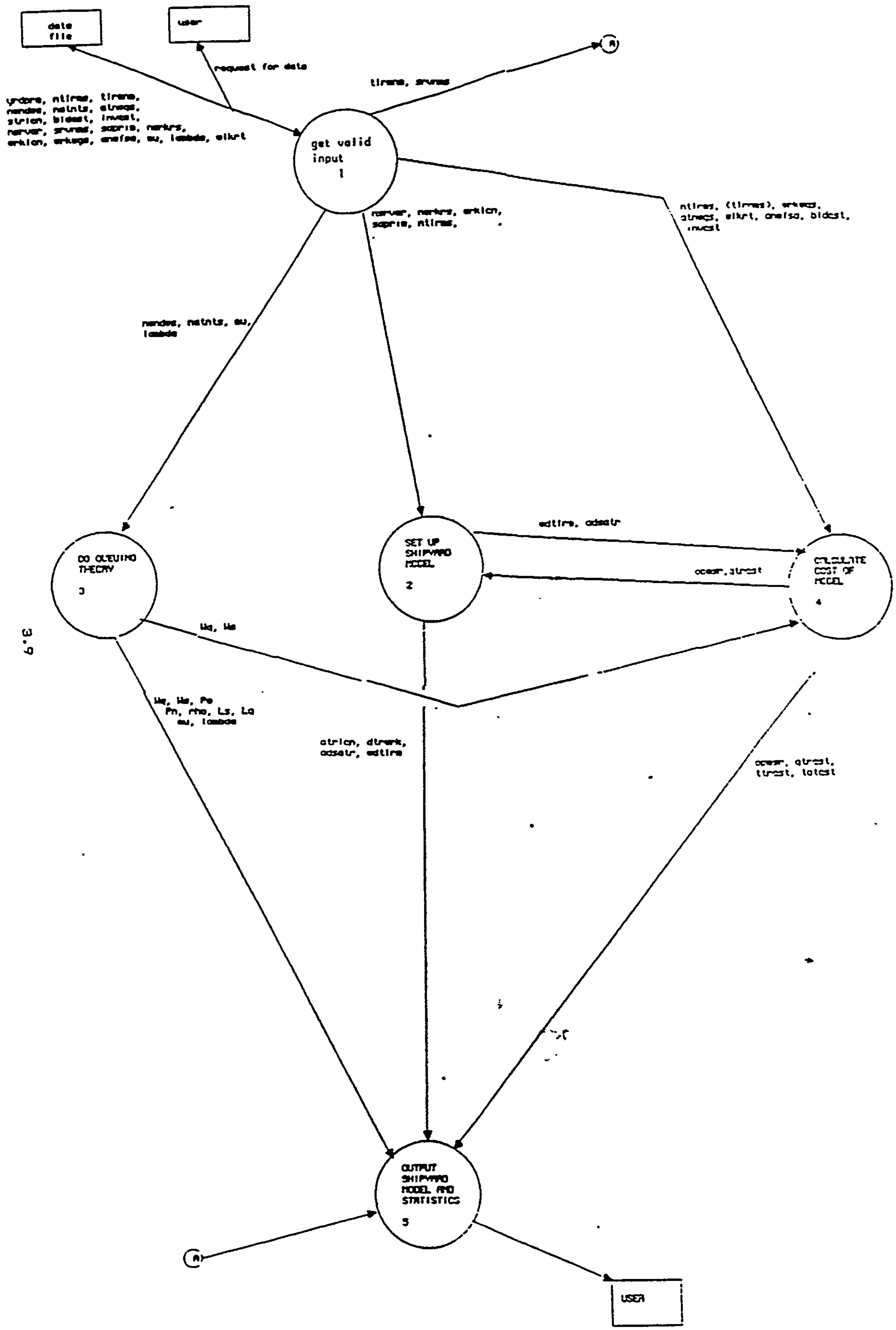
3.5.4 Queueing Information

A queueing study should be performed for each toolroom, This is most easily done by the attendant and a stopwatch. The attendant needs to record the following: (1) Arrival Time at the Toolroom (2) Arrival Time at the Window (3) Departure Time from the window (4) (origin of the Customer. Again, if all of the customers who enter the tool room can be recorded, the queuing study and number of customers being serviced can be done at the same time. Once enough data is processed, the average number of persons arriving at the window per minute can be determined for the model. The model assumes that arrivals are randomly distributed, which characterizes a Poisson distribution. Hence, the average arrival rate and mean service rate are the only rates needing to be calculated.

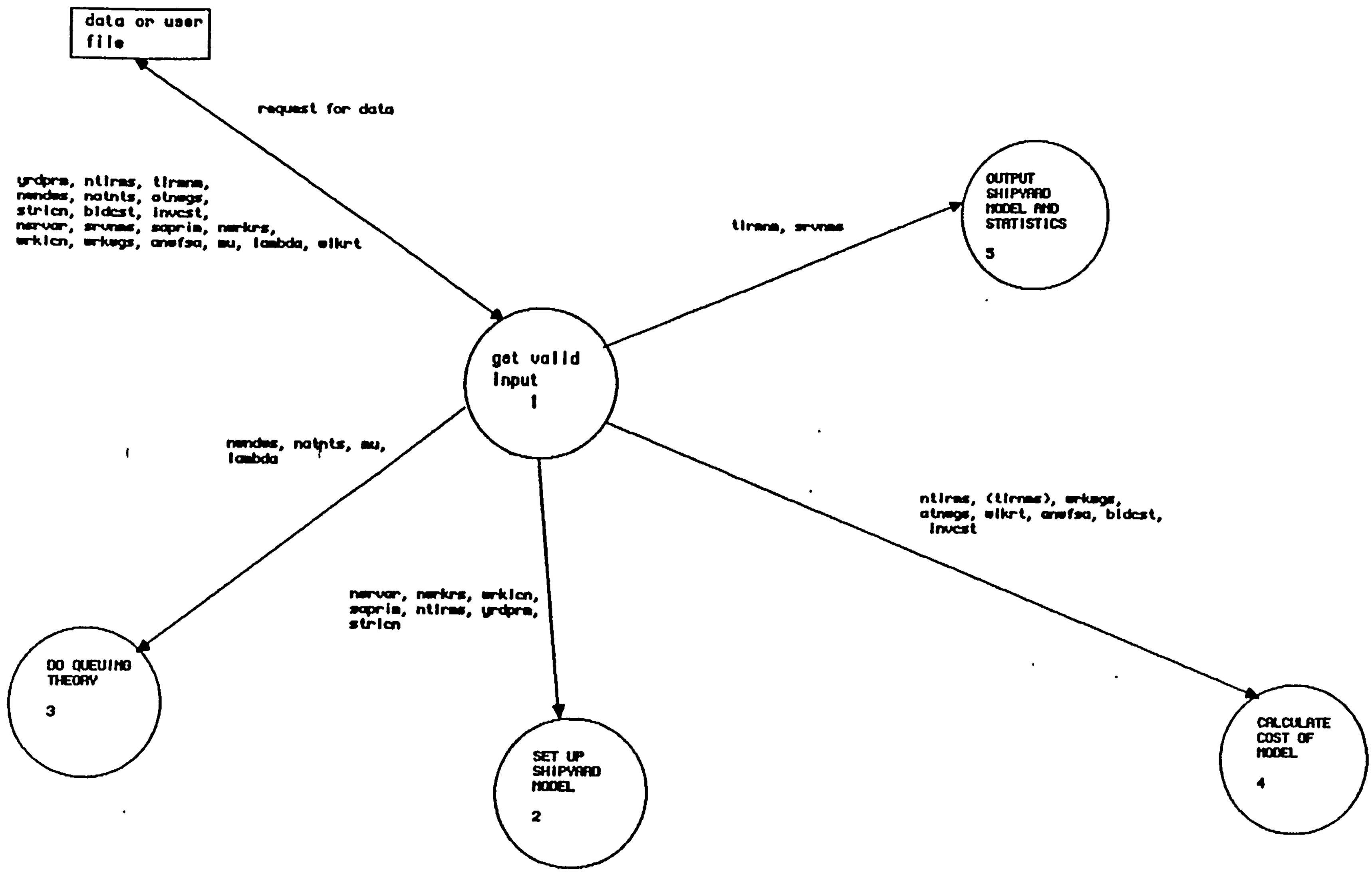
Appendix A

Data Flow Diagrams

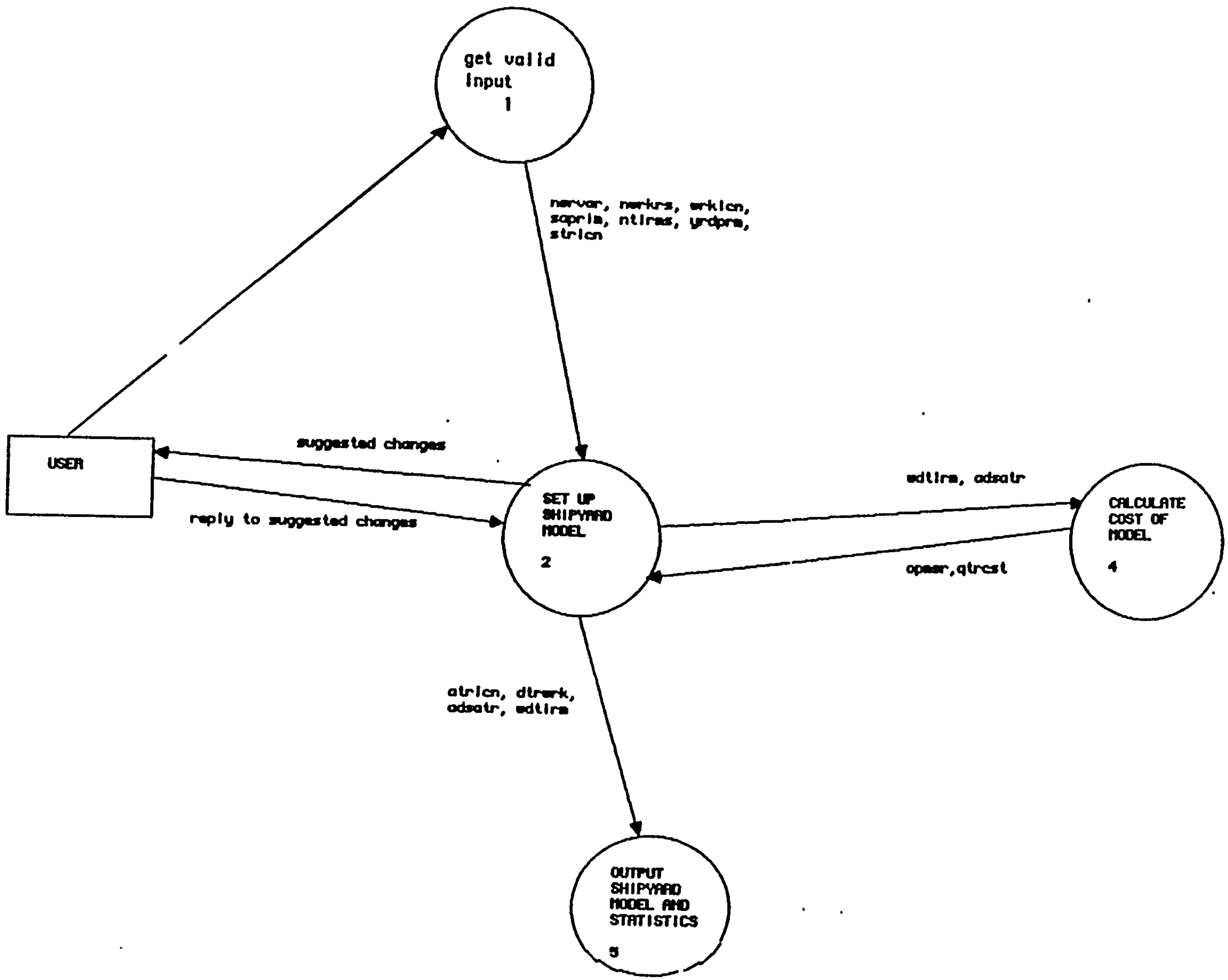
Data Dictionary


user
get valid input 1
DO QUEUING THEORY 3
SET UP SHIPYARD MODEL 2
4
Wq, Ws
OUTPUT SHIPYARD MODEL AND STATISTICS 5
USER

Process 'Get Valid Input' data flow diagram

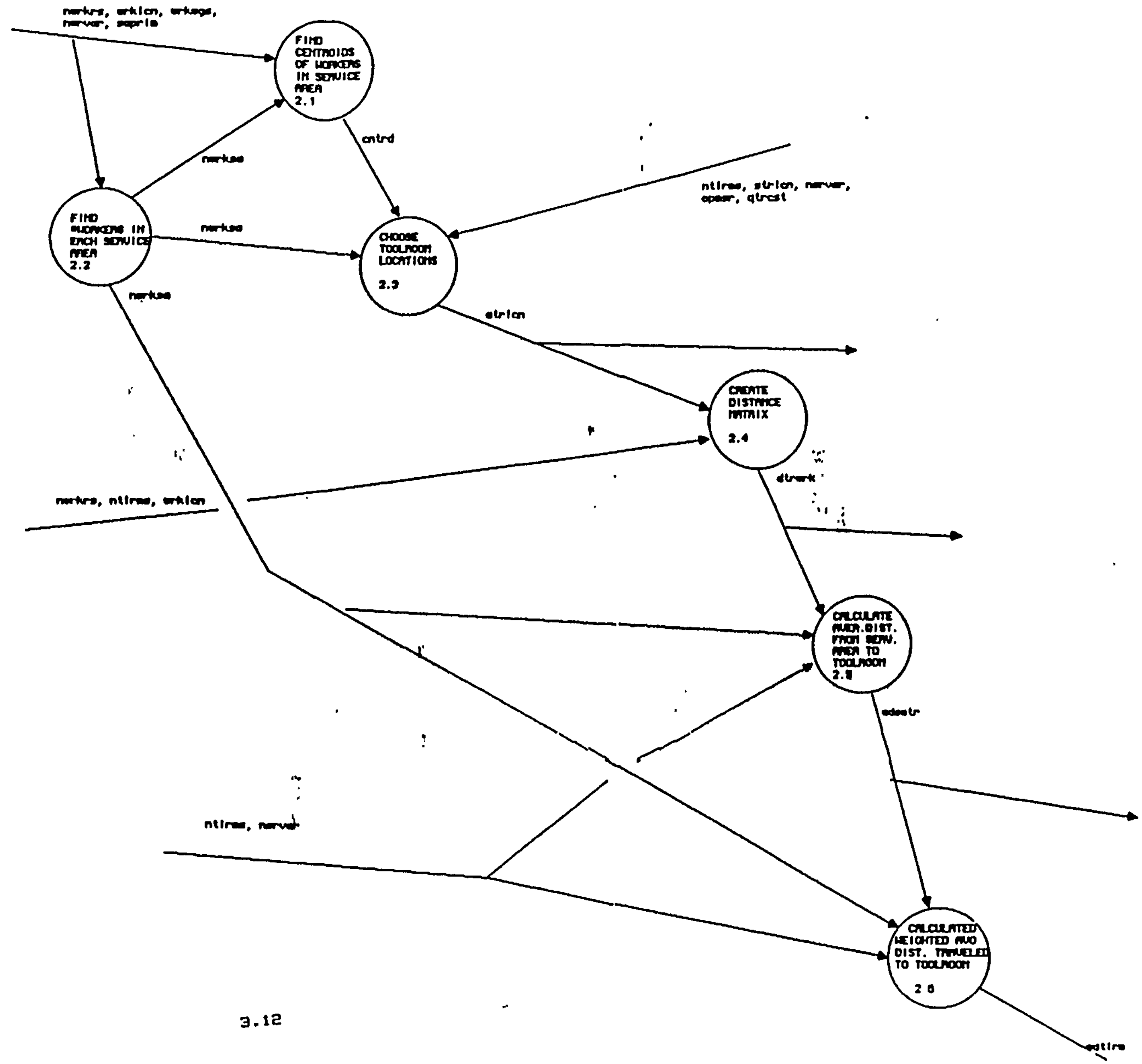


Process 'Set up Shipyard Model' data flow diagram



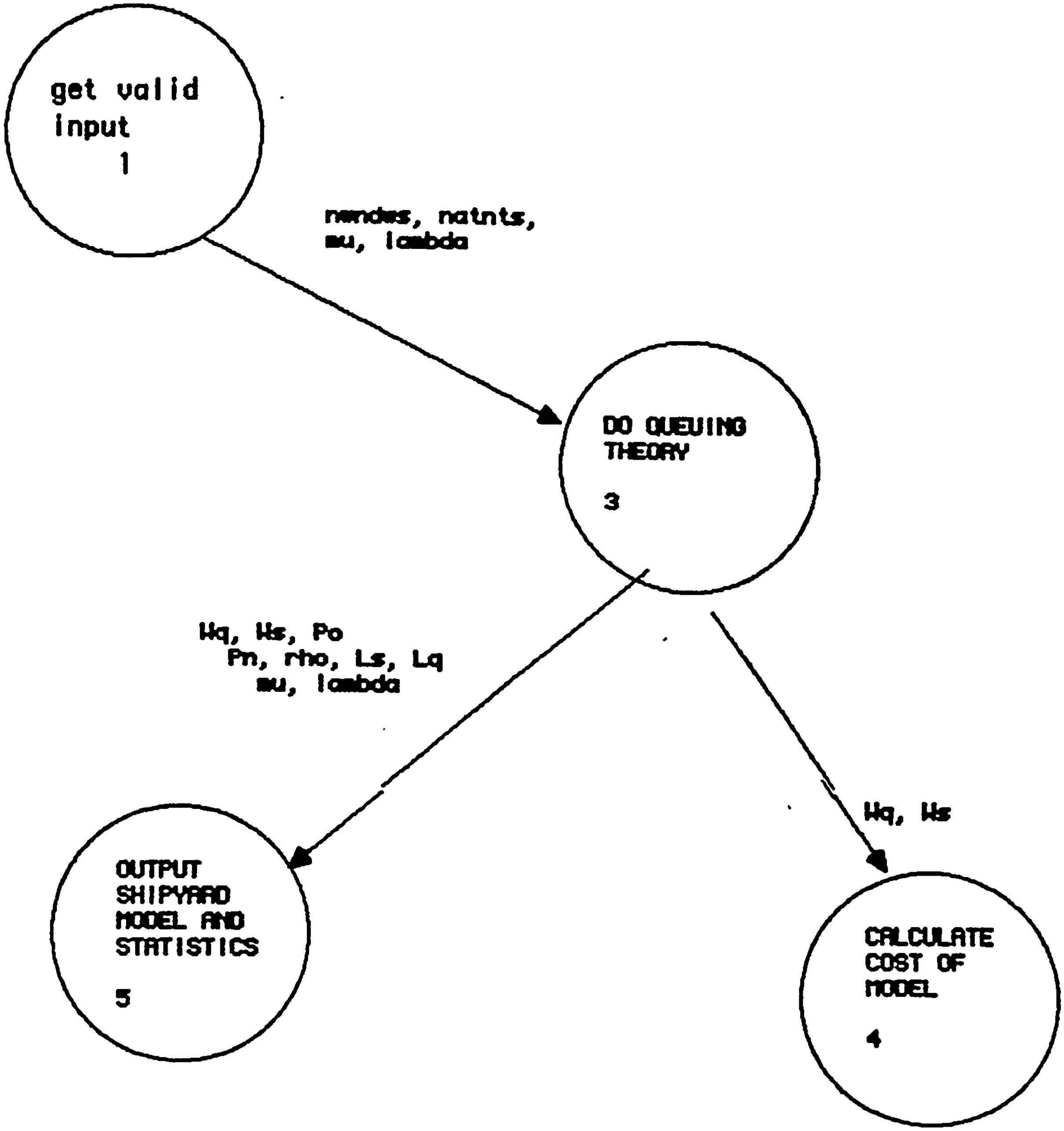
3.11

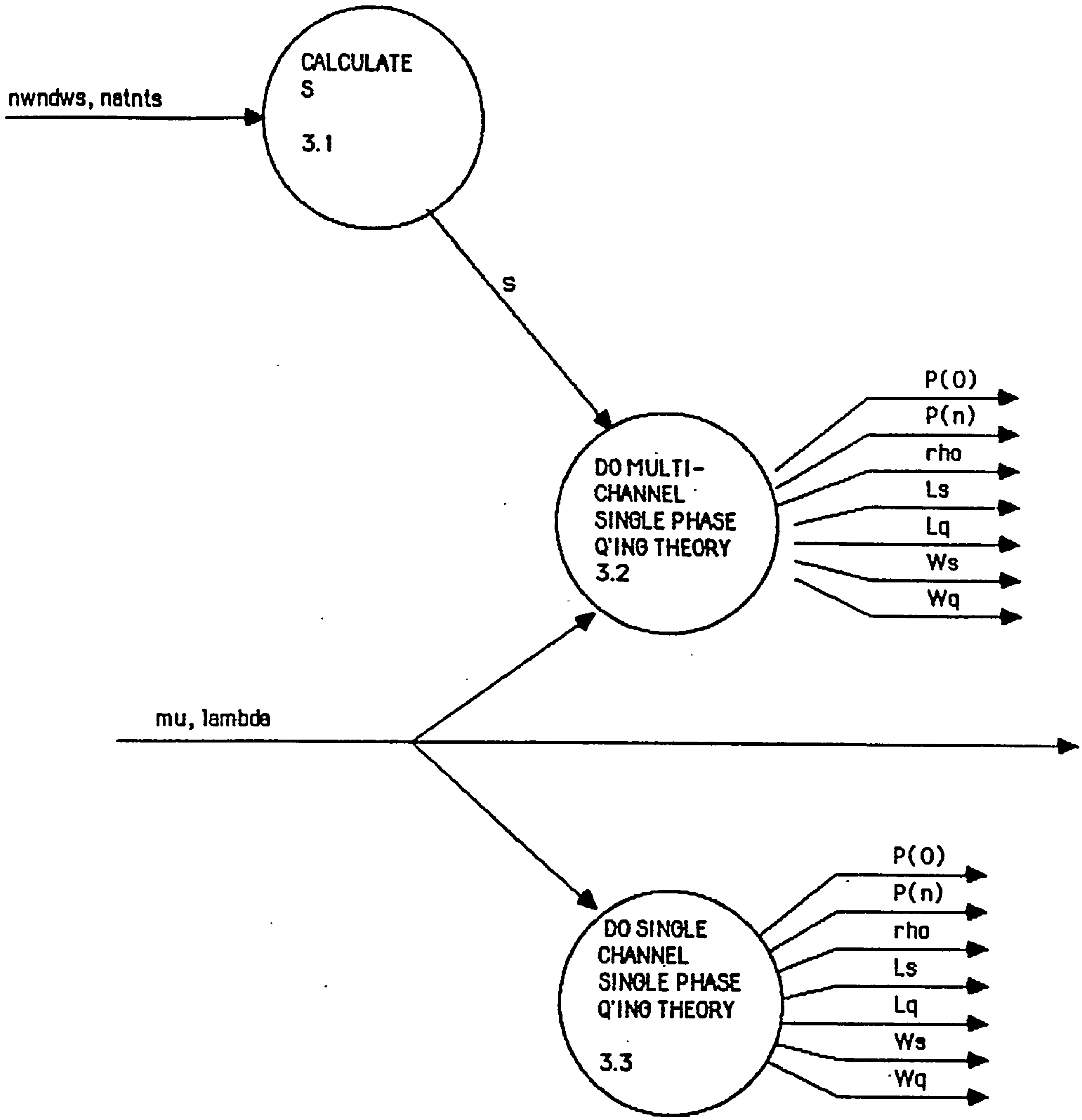
data flow decomposition of Process #2



3.12

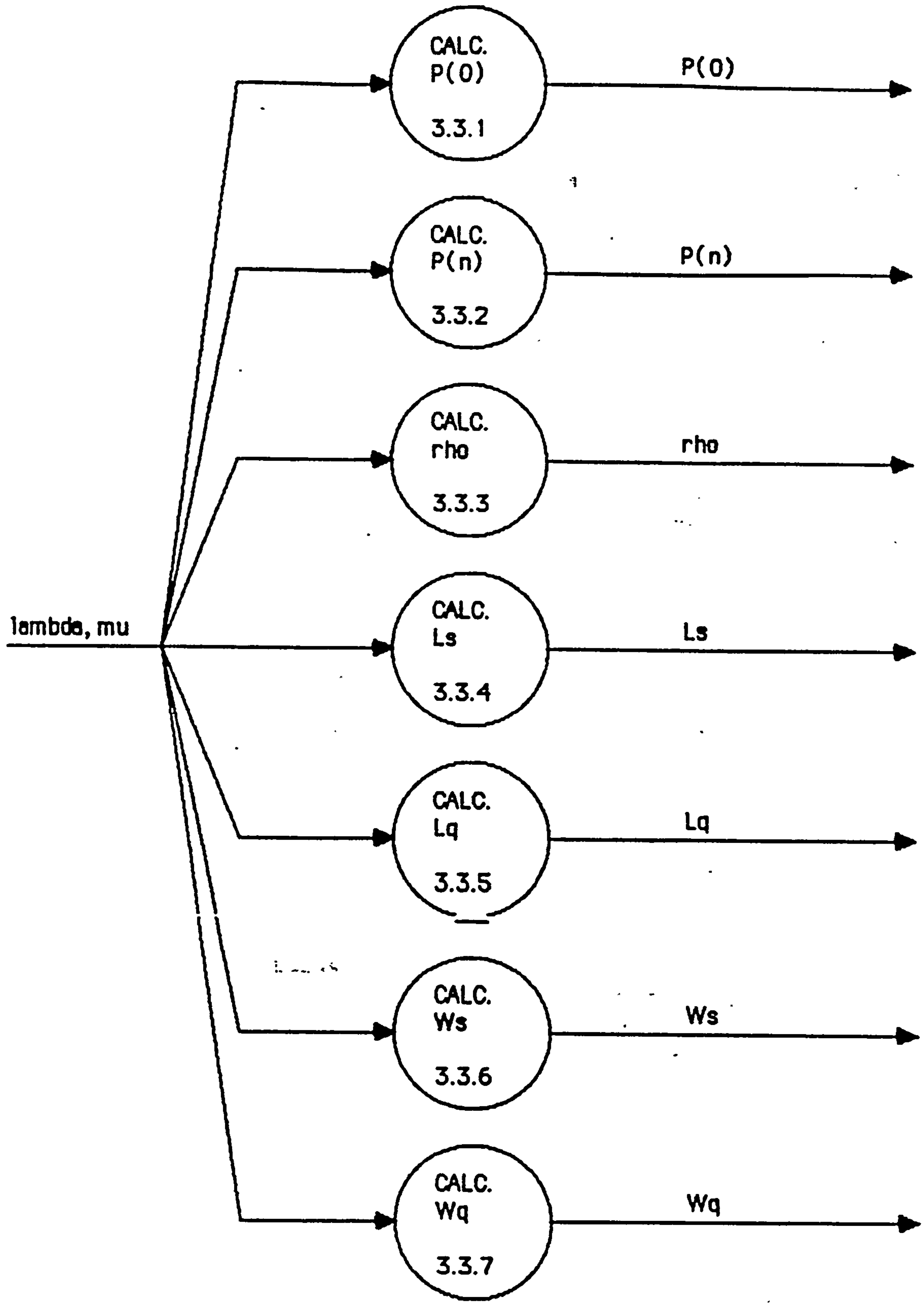
Process 'Do Queuing Theory' data flow diagram

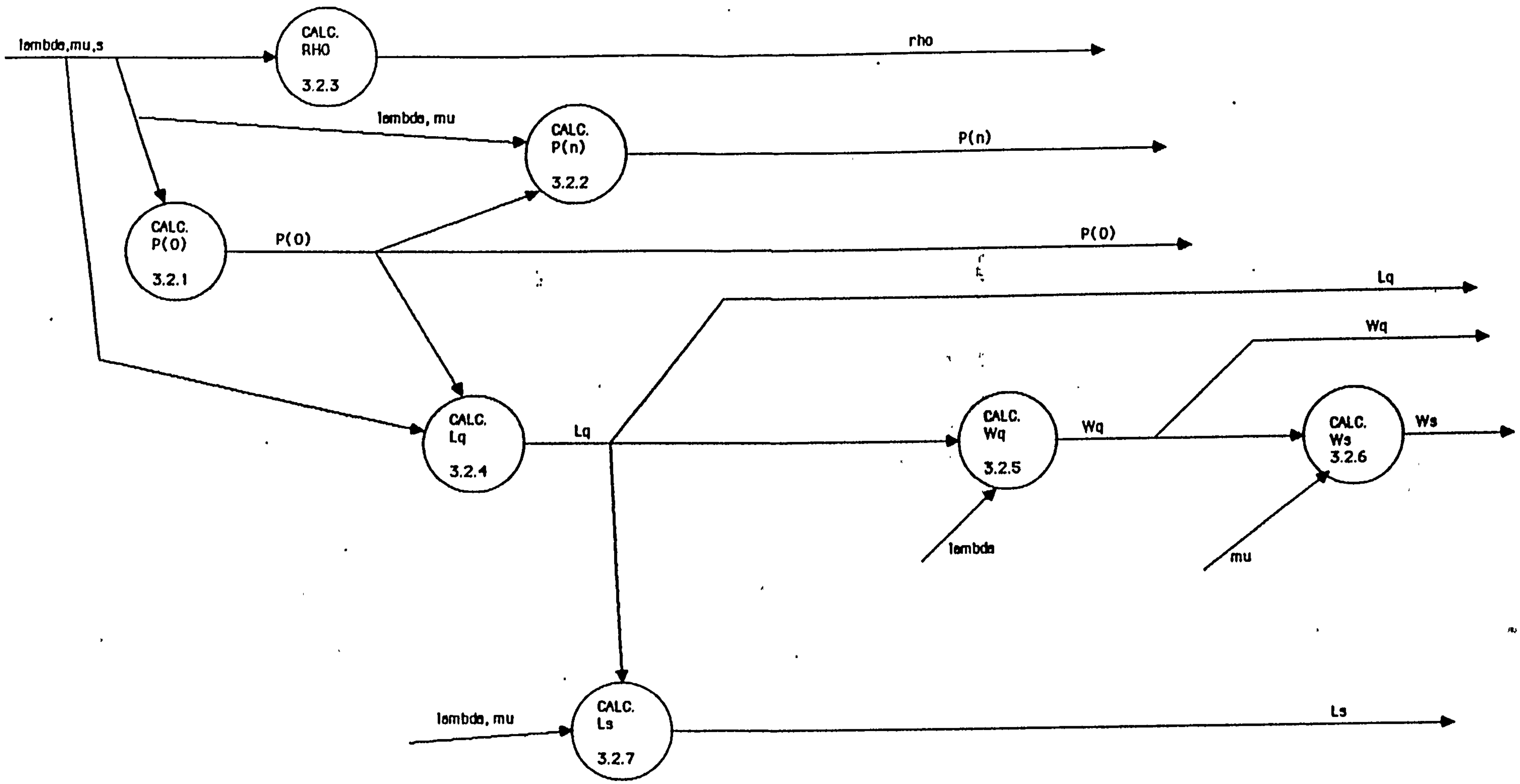




nwndws, natnts
CALCULATE S
3.1
s
DO MULTI-CHANNEL SINGLE PHASE Q'ING THEORY
3.2
P(0)
P(n)
rho
Ls
Lq
Ws
Wq
mu, lambda
DO SINGLE CHANNEL SINGLE PHASE Q'ING THEORY
3.3
P(0)
P(n)
rho
Ls
Lq
Ws
Wq

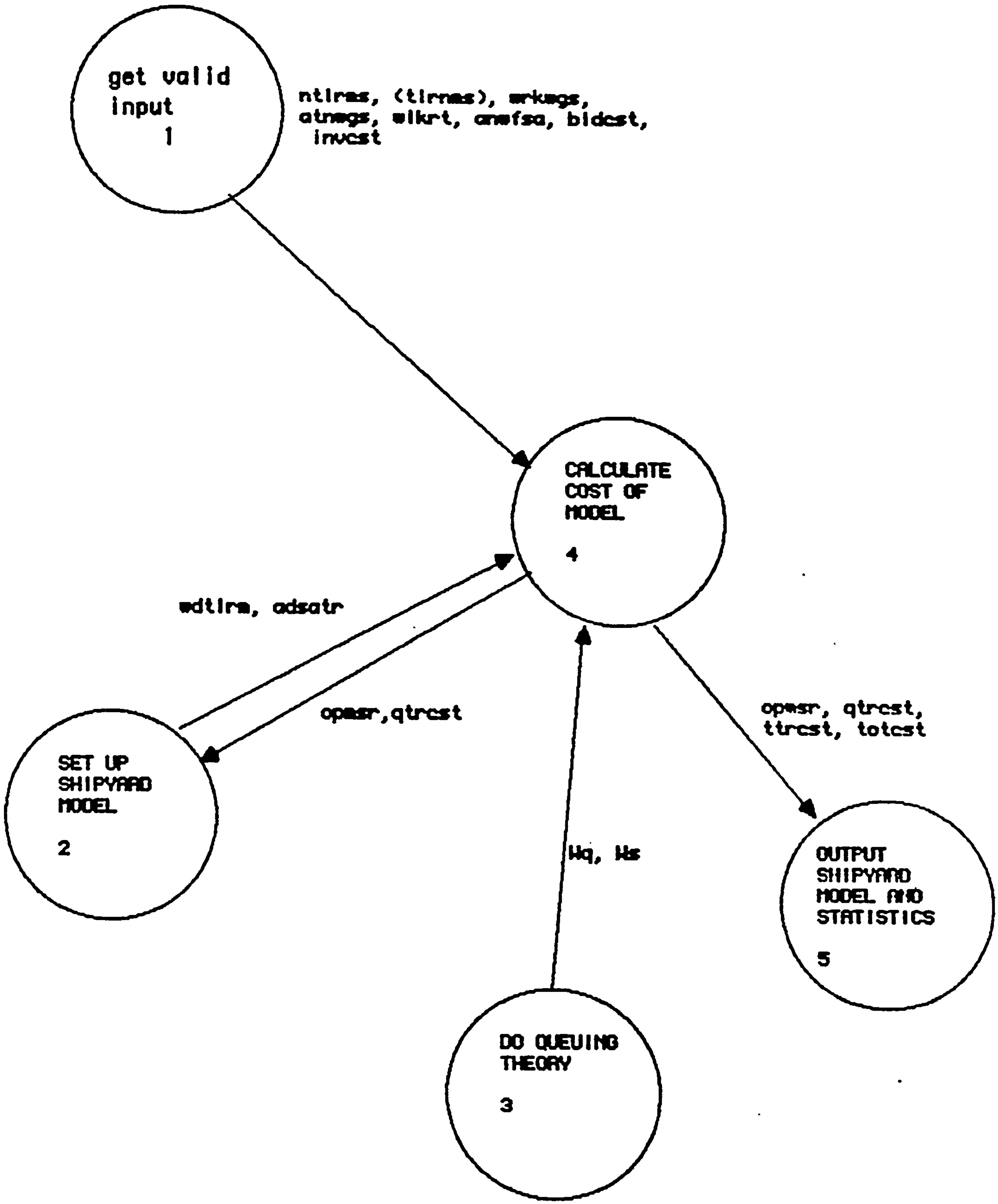
data flow decomposition of Process #3.3

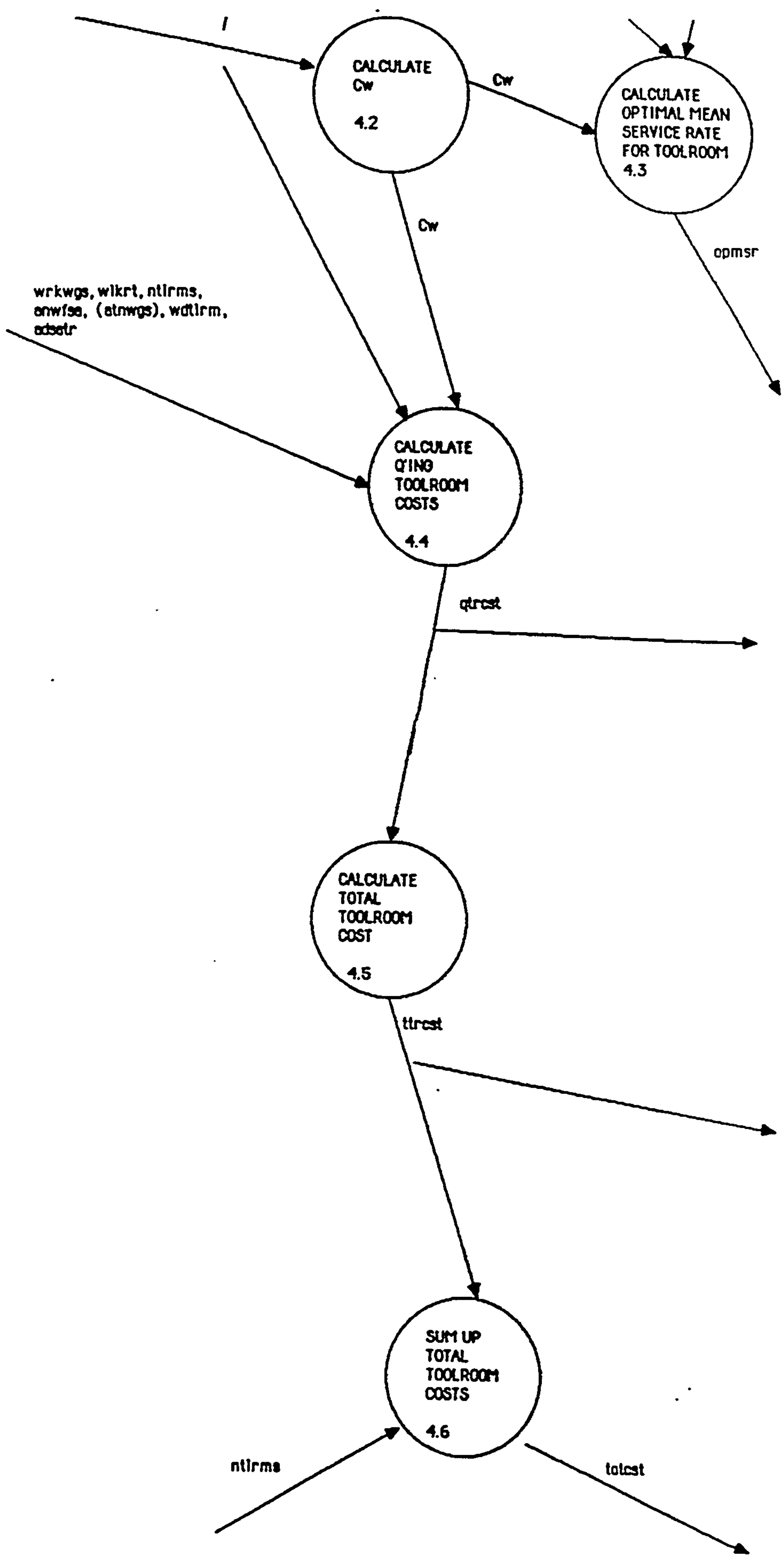



lambda,mu,s
CALC.
RHO
3.2.3
rho
lambda, mu
CALC.
P(n)
3.2.2
P(n)
CALC.
P(0)
3.2.1
P(0)
P(0)
Lq
Wq
CALC.
Lq
3.2.4
Lq
CALC.
Wq
3.2.5
Wq
CALC.
Ws
3.2.6
Ws
lambda
mu
lambda, mu
CALC.
Ls
3.2.7
Ls

3.16

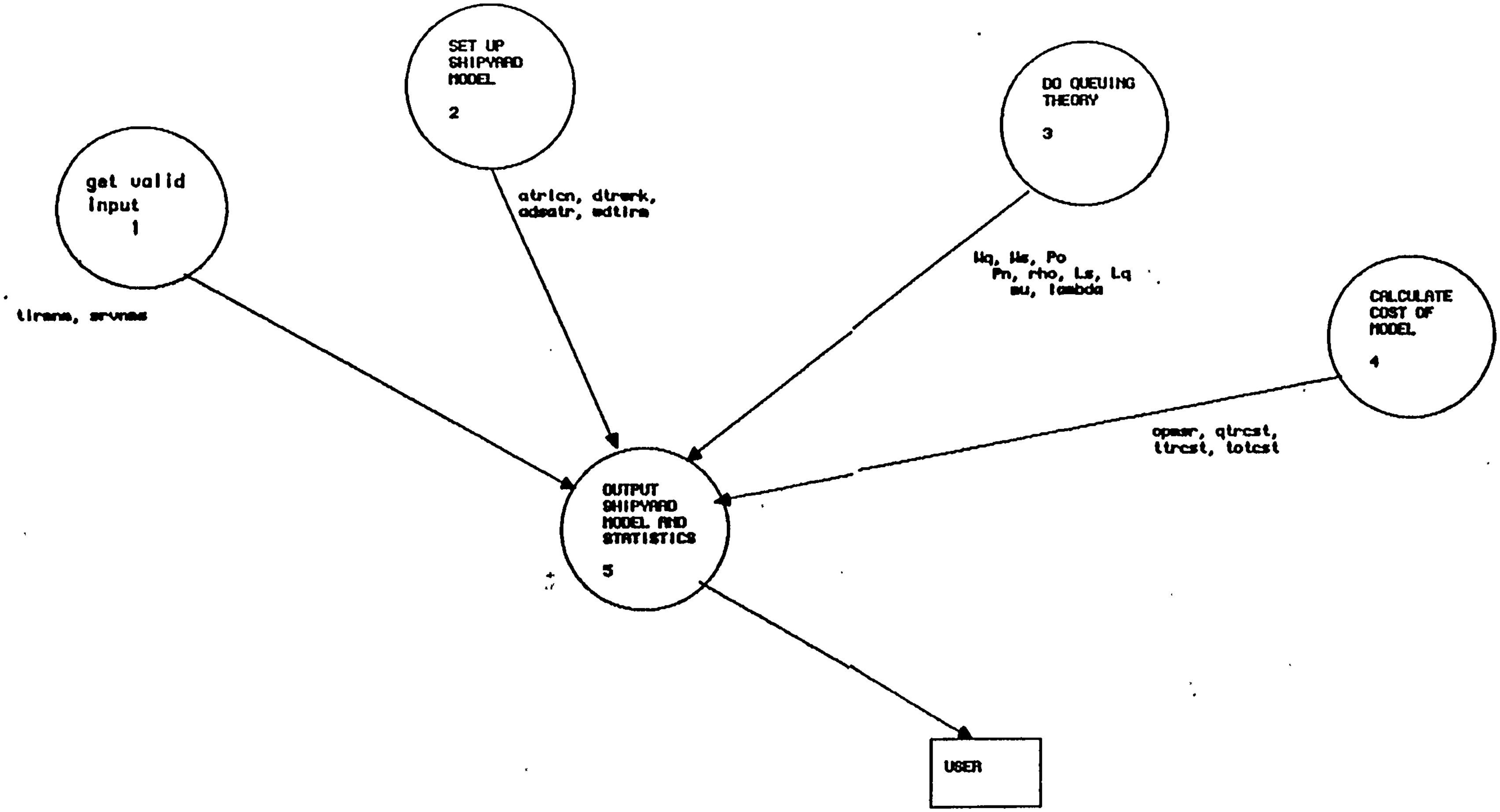
Process 'Calculate Cost of Model' data flow diagram





CALCULATE Cw
4.2
Cw
CALCULATE OPTIMAL MEAN SERVICE RATE FOR TOOLROOM
4.3
opmsr
Cw
wrkwgs, wlkrt, ntlrms, anwfse, (atnwgs), wdtlrm, adsetr
CALCULATE Q'ING TOOLROOM COSTS
4.4
qtrcst
CALCULATE TOTAL TOOLROOM COST
4.5
ttrcst
SUM UP TOTAL TOOLROOM COSTS
4.6
ntlrms
totcst

Process 'Output Shipyard Model and Statistics' data flow diagram



DATA DICTIONARY

* Input from user or file

<u>Toolroom Data</u>

* ntlrms	#toolrooms
* tlrnm	toolroom name
* nwndws	#windows
* natnts	#attendants
* mu	mean service rate for a toolroom
* lambda	mean arrival rate for a toolroom
* strlcn*	*suggested toolroom locations
atrlcn	actual toolroom location
ptrlcn	'perfect' toolroom location
* bldcst	building cost
* invcst	inventory cost
* anwfsa	'average # workers that use toolroom from particular service area'
* atnwgs	attendants' wages

<u>Shipyard Data</u>

* yrdprm*	*shipyard perimeters

<u>Worker Data</u>

* nwrkrs	#workers
* wrklcn	workers' locations
* wrkwgs	workers' wages
wrksa	workers' service area
* wlkrt	walk rate ft/min

<u>Service Area Data</u>

* nsrvar	#service areas
* srvnms	service areas' names
* saprim	service area perimeters
nwrksa	#workers in a service areas
cntrd	'centroid' in a service area

Toolroom Statistics

Wq	mean time spent waiting in line
Ws	mean time spent waiting in system
Po	P(0) probability of empty queue
Pn	P(n) probability of n units in queue
rho	utilization factor
Ls	expected #units in system
Lq	mean length of queue
opmsr	optimal mean service rate
qtrcst	queuing toolroom cost
ttrcst	total toolroom cost
Cw	waiting cost per unit
Cs	servicing cost per unit

Other Data

dtrwrk	distances between toolrooms and workers
adsatr	average distance from service areas to toolrooms
wdtlrm	weighted average distance to each toolroom
totcst	cost for all toolrooms
s	# of parallel channels

statistical history display of all statistics and data to make sense of statistics for each iteration

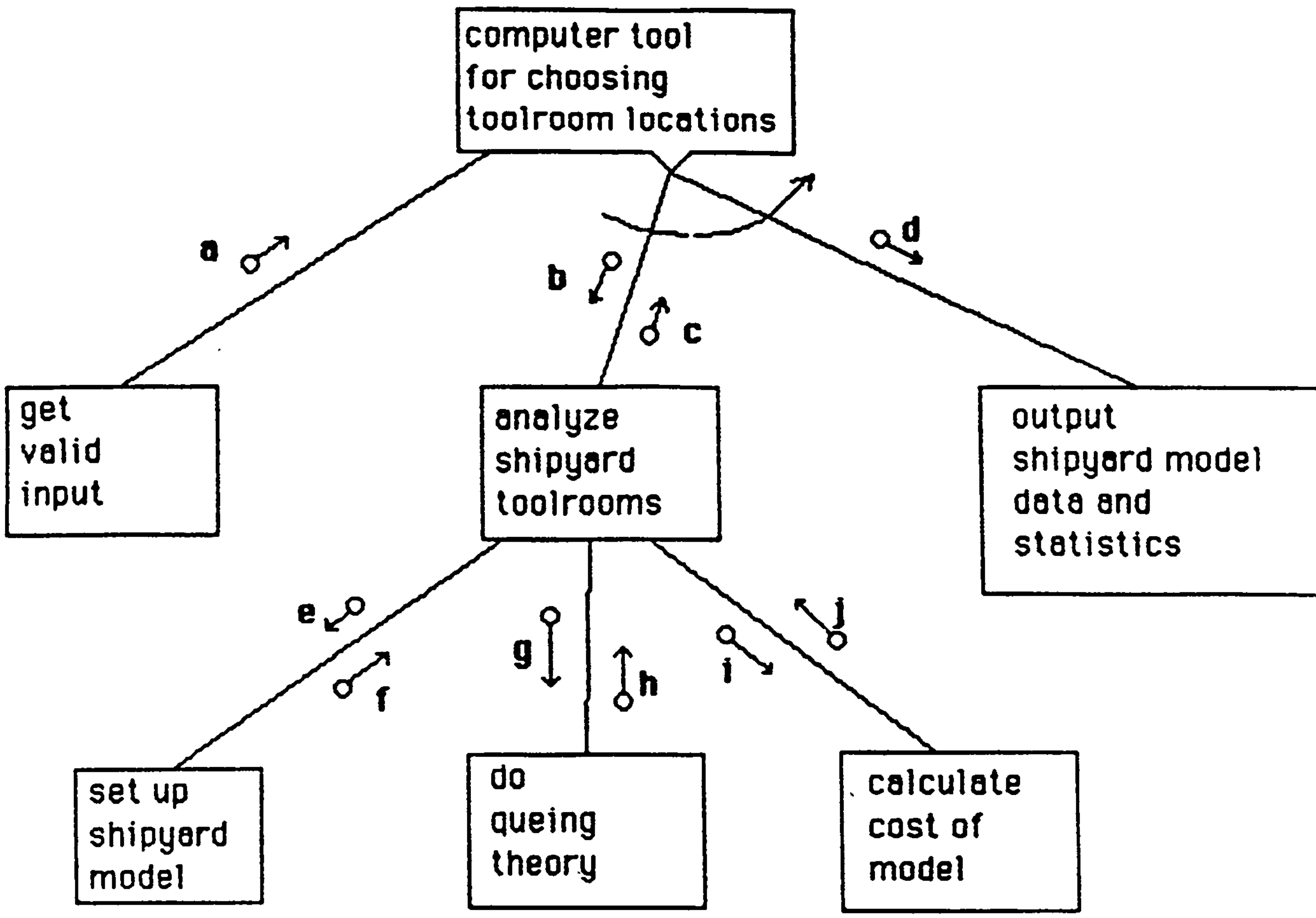
d	noticeable differences
stc	suggested toolroom changes

Appendix B

Structured Charts

structure chart #1

control of flow for main program



structure chart

ntlrms, tlrnm, nwndws, natnts, mu, lambda, strlcn, bldcst, lnvcst, anwfsa, atnwgs, yrdprm, nwrkrs, wrklcn, wrkwgs, nsrvar, srvnms, saprim, wlkrt

nwndws, natnts, mu, lambda, nwrkrs, wrklcn, saprim, ntlrms, yrdprm, strlcn, wrkwgs, atnwgs, wlkrt, anwfsa, bldcst, lnvcst, tlrnm

Wq, Ws, Po, Pn, rho, Ls, Lq, mu, lambda, atrlcn, dtrwrk, adsatr, wdtlrm, opmsr, qtrcst, ttrcst, totcst

{ c U { tlrnm, srvnms }}

nwrkrs, wrklcn, saprim, ntlrms, yrdprm, strlcn, opmsr, qtrcst, nsrvar

wdtlrm, adsatr, atrlcn, dtrwrk

nwndws, natnts, mu, lambda, ntlrms

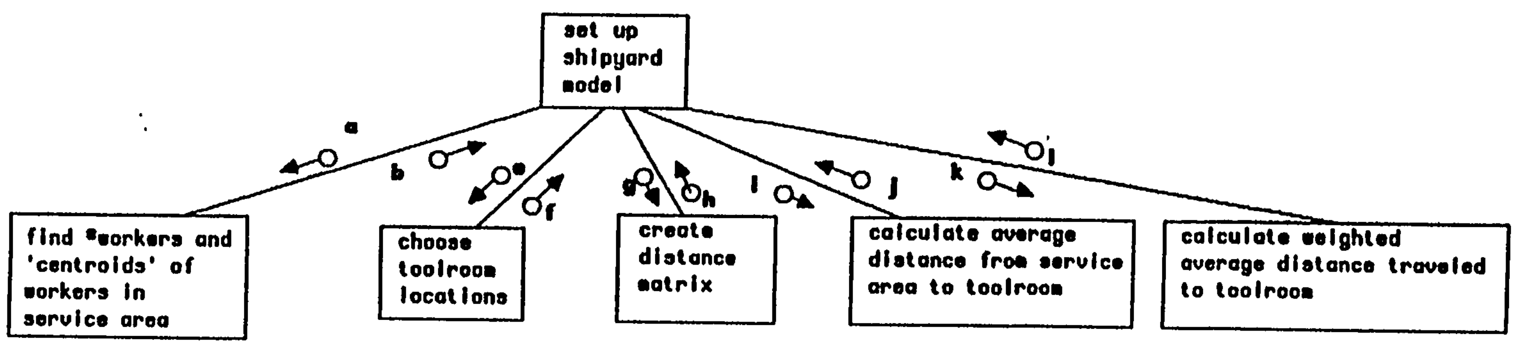
Wq, Ws, Po, Pn, rho, Ls, Lq

wdtlrm, adsatr, Wq, Ws, ntlrms, wrkwgs, atnwgs, wlkrt, anwfsa, bldcst, lnvcst

msrvrt, qtrcst, ttrcst, totcst

STRUCTURE CHART #2

'Set Up Shipyard Model' module control of flow



3.25

structure chart *2
data description

a: { nwrkrs, wrklcn, nsrvar, saprim }

b: { nwrksa,cntrd }

e: {nwrksa, cntrd, ntlrms, nsrvar, opmsr, qtrcst }

f: { atrlcn }

g: { f U { nwrkrs, wrklcn, ntlrms }}

h: { dtrwrk }

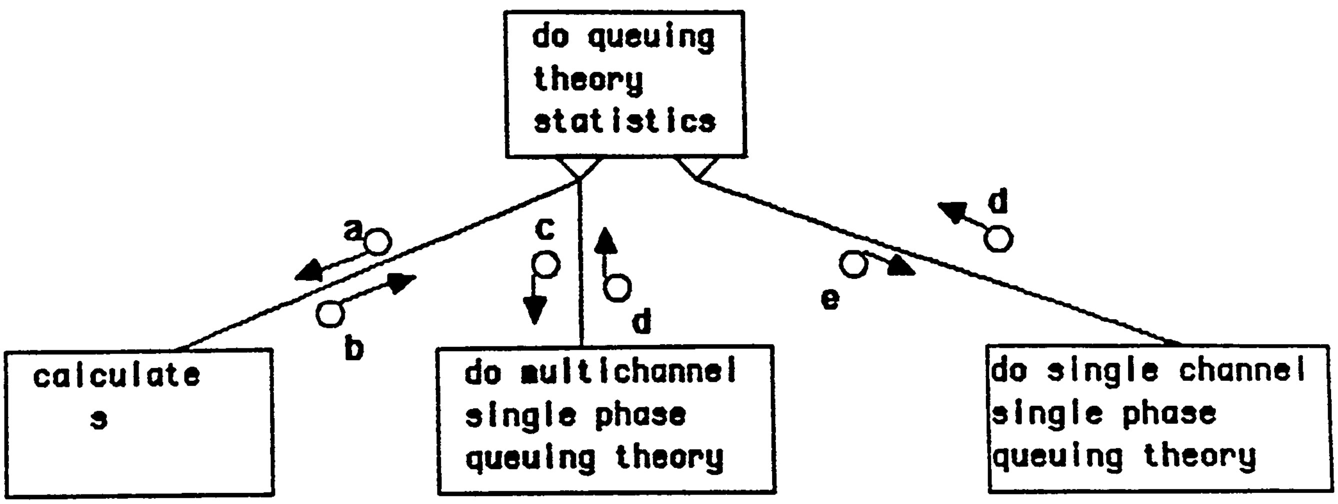
i: {h U { ntlrms, nsrvar, nwrksa }}

j: { adsatr }

k: {j U { ntlrms, nsrvar. nwrksa } }

l: { wdtlrm }

STRUCTURE CHART # 3



a: { nmndms, natnts }

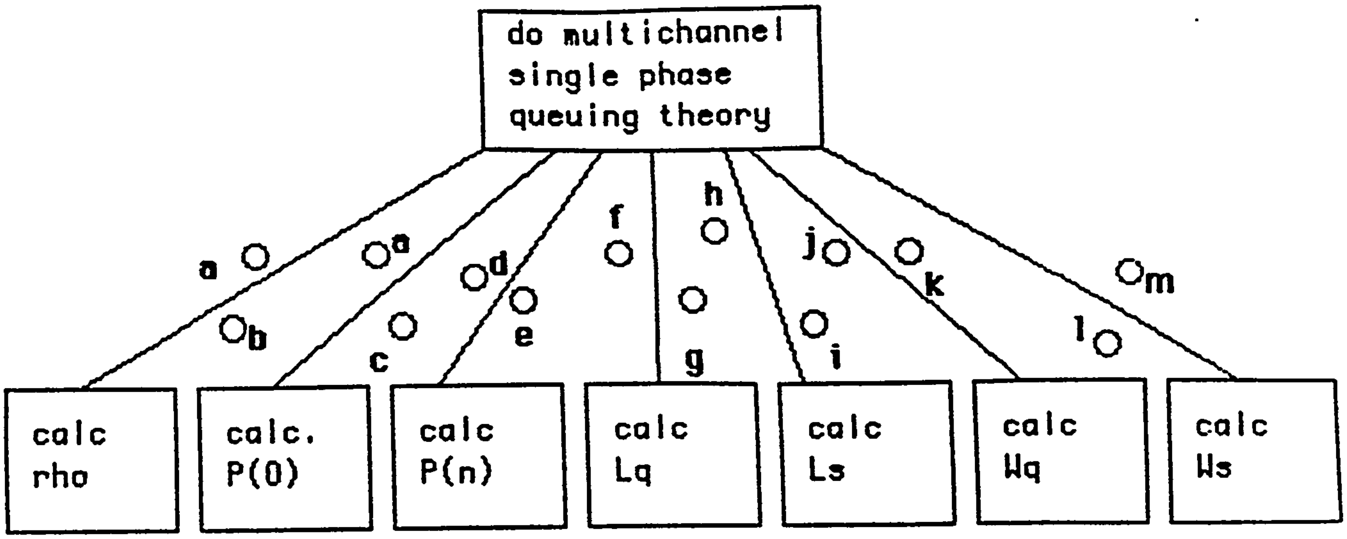
b: { s }

c: {b U { mu, lambda }}

d: { Po, Pn, rho, Ls, Lq, Ws, Wq }

e: { c \ b }

STRUCTURE CHART *4



all are inline

a: { lambda, mu, s }

b: { rho }

c: { Po }

d: { a U { lambda, mu } }

e: { Pn }

f: { a U c }

g: { Lq }

h: { g U { lambda, mu } }

i: { Ls }

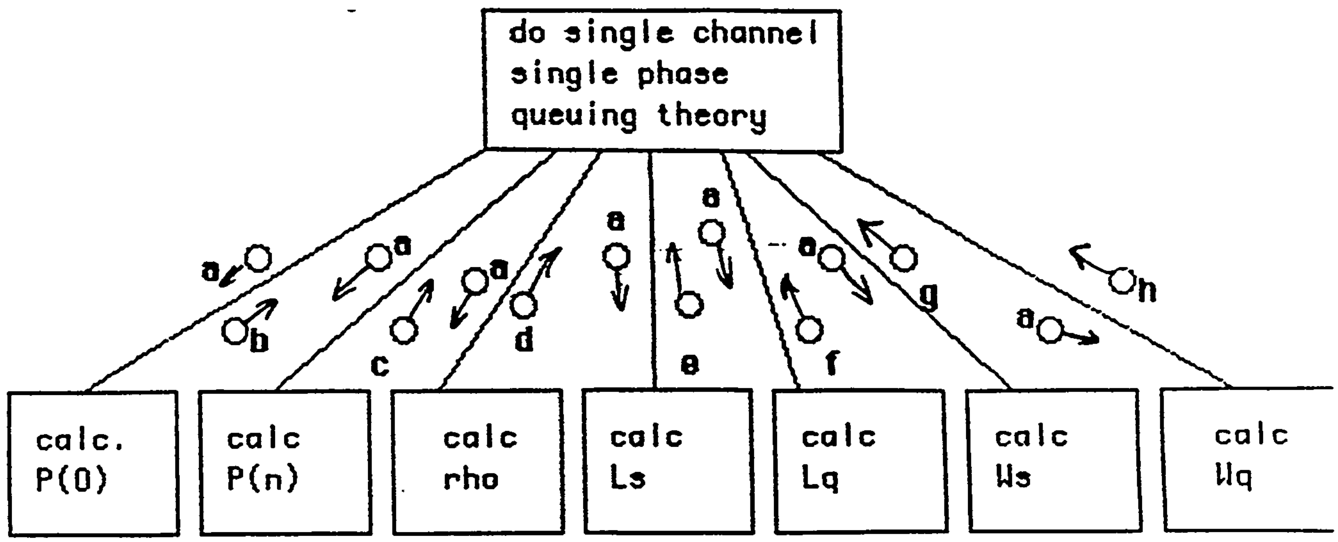
j: { g U { lambda, mu } }

k: { Wq }

l: { k U { mu } }

m: { Ws }

STRUCTURE CHART #5



all are inline

a: { lambda, mu }

b: { Po }

c: { Pn }

d: { rho }

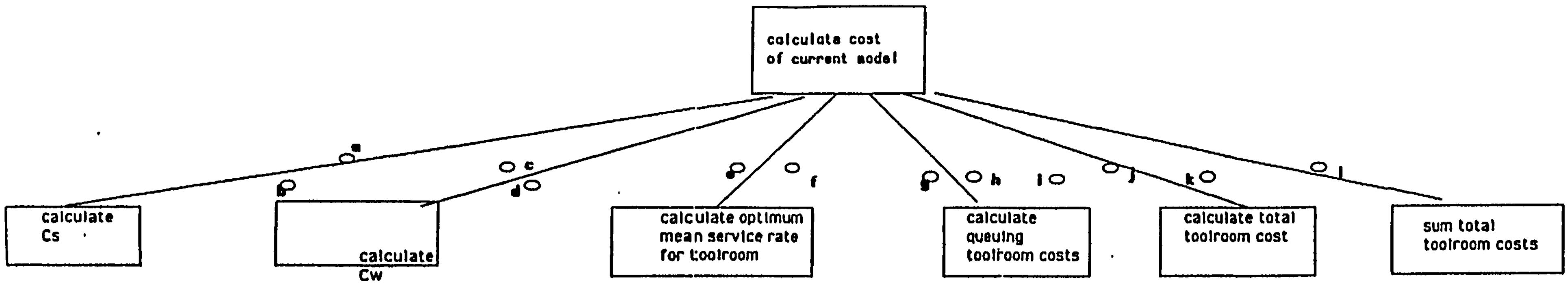
e: { Ls }

f: { Lq }

g: { Ws }

h: { Wq }

STRUCTURE CHART - 6



structure chart #6
data description

a: { Ws, Wq, ntlrms, wrkwgs, atnwgs }

b: { Cs }

c: { Wq, ntlrms, wrkwgs }

d: { Cw }

e: { d U b U { ntlrms } }

f: { opmsr }

g: { d U b U { ntlrms, wrkwgs, wlkrt, anwfsa, wdtlrm, adsatr } }

h: { qtrcst }

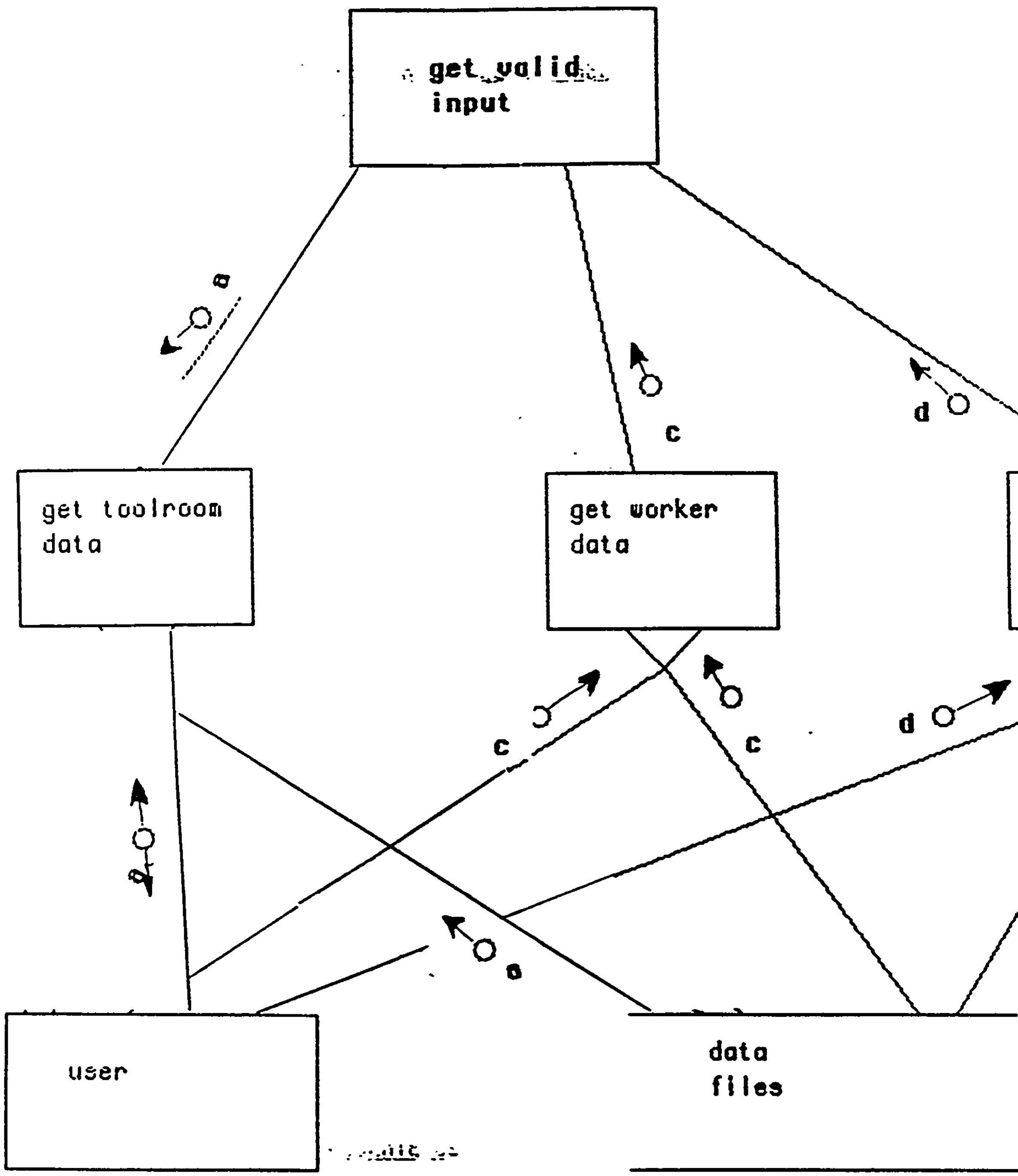
i: { h U { ntlrms, bldcst, invcst } }

j: { ttrcst }

k: { j U { ntlrms } }

l: { totcst }

structure chart 3.7



Appendix C

Process Specification

PROCESS SPECIFICATIONS

1 GET VALID INPUT

ask the user to input the information that is designated as input in the data dictionary. e.g. toolroom data, service area data, etc.

2.1 FIND CENTROIDS OF WORKERS IN SERVICE AREAS

2.2 FIND # WORKERS IN EACH SERVICE AREA (combined)

```
for each worker loop
  for each service area loop
    if worker is in service area then
        increment x-coord location of workers
        increment y-coord location of workers
        increment counter of workers in identified service
          area
        increment number of workers in service areas
    end if
  end loop
end loop
x-coord of service area's centroid is the sum of worker's
    x-coord location divided by # workers in service area
y-coord of service area's centroid is the sum of worker's
    y-coord location divided by # workers in service area
```

2.3.1 CALCULATE 'PERFECT' TOOLROOMS LOCATIONS

```
IF #service areas == #toolrooms
THEN
    assign locations of toolrooms as of service areas
        centroids
ELSE IF #service areas > #toolrooms
    sort service areas - heaviest first
    loop until toolrooms left = 1
        partition weight of service areas based on the
           number of toolrooms left
        assign a toolroom to the first set of service areas
           whose total weight is equal to partioned weight
           toolroom is located at the centroid of the
           set of service areas
    end loop
    for last toolroom assign toolroom's location to the
        centroid of the set of remaining service areas
ELSE IF #service areas < #toolrooms
    assign to each service area's centroid a toolroom
    for allocating the remaining toolrooms follow the
    algorithm used when the #service areas > # toolrooms
END IF
```

2.4 CREATE DISTANCE MATRIX

```
FOR each toolroom LOOP
    FOR each worker LOOP
        calculate distance between worker and toolroom
        using following formula:
        distance =
           sqrt((workers' x - toolrooms' x)**2
              + ( (workers' y - toolrooms' y)**2)
    END LOOP
END LOOP
```

2.5 CALCULATE AVERAGE DISTANCE FROM SERVICE AREAS TO TOOLROOMS

```
FOR each toolroom LOOP
    FOR each service area LOOP
        use distance formula to calculate distance between
        centroid of service area and toolroom location
    end loop
end loop
```

```
2.6 CALCULATE WEIGHTED AVERAGE DISTANCE TRAVELED TO
    TOOLROOM
    FOR each toolroom LOOP
        FOR each service area LOOP
            calculate the following formula:
            (#workers in a service area/#workers)
            * average distance from service area to toolroom
        END LOOP
    END LOOP

3.1 CALCULATE S
    IF #attendants > #windows
    THEN
        s := #windows
    ELSE
        s := #attendants
    END IF
```

CALCULATIONS FOR MULTICHANNEL QUEUING THEORY

3.2.1

$$P(0) = \frac{1}{\left(\sum_{n=0}^{s-1} \frac{(\lambda/\mu)^n}{n!}\right) + \frac{(\lambda/\mu)^s}{s!}\left(1 - \frac{\lambda}{s\mu}\right)^{-1}}$$

3.2.2

$$P(n) = \frac{(\lambda/\mu)^n}{n!} P(0) \quad ,0 \leq n \leq s$$

$$= \frac{(\lambda/\mu)^n}{s!\,s^{n-s}} P(0) \quad ,n \geq s$$

3.2.3

$$\rho = \lambda / s\mu$$

3.2.4

$$L_q = \frac{P(0)(\lambda/\mu)\, e^{s}}{s!\,(1-e)^2}$$

3.2.5

$$W_q = L_q / \lambda$$

3.2.6

$$W_s = W_q + (1/\mu)$$

3.2.7

$$L_s = L_q + (\lambda/\mu)$$

CALCULATIONS FOR SINGLE CHANNEL

3.3.1

$$P(0) = 1 - (\lambda/\mu)$$

3.3.2

$$P(n) = P(0)(\lambda/\mu)^n$$

3.3.3

$$\rho = \lambda/\mu$$

3.3.4

$$L_s = \lambda/(\mu - \lambda)$$

3.3.5

$$L_q = \lambda^2 / [\mu(\mu - \lambda)]$$

3.3.6

$$W_s = 1/(\mu - \lambda)$$

3.3.7

$$W_q = \lambda / [\mu(\mu - \lambda)]$$

4.1 CALCULATE Cs

Cs = (Ws - Wq) * (attendants' wage + workers' wage)

4.2 CALCULATE Cw

```
Cw = Wq * (workers' wage)
```

4.3 CALCULATE OPTIMUM MEAN SERVICE RATE FOR TOOLROOM

```
opt. mean = lambda + sqrroot(lambda*Cw/Cs)
```

4.4 CALCULATE TOOLROOM COSTS DUE TO QUEING

```
FOR each toolroom LOOP
    FOR each service area LOOP
        cost for 1 worker to walk to toolroom =
        2*avg.dist to toolroom (ft) /
        walkrate (ft/min) * wage/hr/60 min./hr

        total cost = total cost +
          #workers from service area * cost for 1 worker..
    END LOOP
END LOOP
```

4.5 CALCULATE TOTAL TOOLROOM COSTS

```
FOR each toolroom LOOP
    queing cost + building cost + inventory cost
END LOOP
```

4.6 CALCULATE COST FOR ALL TOOLROOMS

```
FOR 1..#toolrooms LOOP
    total toolroom cost := total toolroom cost +
        toolroom(i) cost
END LOOP
```

5 OUTPUT SHIPYARD MODEL DATA AND STATISTICS

```
output all statistics with appropriate data to
enable user to understand statistics
query user for finish
IF finish THEN
    quit
ELSE
    perform another iteration
END IF
```

Appendix D

Final Code

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC(

 COMPUTER TOOL FOR CHOOSING TOOLROOM LOCATIONS
PROGRAM MAIN
 SWITCHES 1 PRINT BANNER + 4 DELETE AFTER PRINTING + 16 PRINT PAGE
   HEADERS + 64 APPEND A FINAL FORM QUEUE
        PROGRAM  MAIN(SWITCHES=85)


INTEGER NTLRMS
CHARACTER*20 TLRMNM(10)
INTEGER NWNDWS( 10),NATNTS(10)
REAL MU(10),LAMBDA(10)
REAL STRLCN(10,2)
REAL BLDCST( 10),INVCST(10)
REAL ANWFSA( 10),ATNWGS(10)
REAL YRDPRM(4)
INTEGER NWRKRS
REAL WRKLCN(100,2)
REAL WRKWGS
REAL WLKRT
INTEGER NSRVAR
CHARACTER*20 SRVNMS(20)
REAL SAPRIM(20,4)
        REAL WQ(l0),WS( 10),P0(10)
REAL PN(50,10)
REAL RHO(l0),LS(10),LQ(10)
REAL ATRLCN(10,2)
REAL DTRWRK(10,100)
REAL ADSATR(20,10)
REAL WDTLRM(10)
REAL OPMSR(10)
REAL QTRCST(10)
REAL TTRCST(10)
REAL TOTCST
REAL CS(10),CW(10)
INTEGER NWRKSA(20)
REAL CNTRD(20,2)
REAL PTRLCN(10,2)
INTEGER NTIMES
INTEGER QUITIT
INTEGER MNUVAR
INTEGER OPTIMES

 DISPLAY ACTIVITY OPTION MENU

       WRITE(*,l)
       FORMAT(////,lX,80('*' ),//,33X,'***CATD***' ,//,80('*'),////)

OPTIMES = 1
        WRITE(*,3)
        FORMAT(lX,'PLEASE ENTER THE APPROPRIATE INTEGER')
WRITE(*,4)
        FORMAT(2X,'TO CHOOSE ONE OF THE FOLLOWING OPTIONS')
WRITE(*,5)
        FORMAT(10X,'1 - TOOLROOM STATISTICS' )
WRITE(*,6)
        FORMAT(10X,'2 - TOOLROOM ECONOMICS')
```

```
WRITE(*,7)
        FORMAT(10X,'3 - OPTIMAL LOCATIONS')
WRITE(*,8)
        FORMAT(1X,'INTEGER',/,1X,'ENTER:')

READ(*,9,ERR=10)MNUVAR
        FORMAT(I4)
       GOTO 11
0      WRITE(*,15)
       GOTO 2
1      CONTINUE

       WRITE(*,13)
3      FORMAT(/)
      IF (MNUVAR.EQ.1)COUNT=COUNT+1
QUITIT = 1
NTIMES = 1
DO 14 I=1,10
  STRLCN(I,1)= -999.
  STRLCN(I,2)= -999.
         SRVNMS(I)=' '
         SRVNMS(I+10)=' '
4       CONTINUE
5      FORMAT(,/,1X,'***ERROR***MISTYPED DATA',/)

 GET VALID INPUT
CALL GVLNPT(NTLRMS,TLRMNM,NWNDWS,NATNTS,MU,LAMBDA,
    +      STRLCN,BLDCST,INVCST,ANWFSA,ATNWGS,YRDPRM,NWRKRS,
    +      WRKLCN,WRKWGS,WLKRT,NSRVAR,SRVNMS,
    +      SAPRIM,MNUVAR,OPTIMES)
 WHILE NOT ( USER REQUESTS ENDING OF PROGRAM )
   ANALYZE SHIPYARD TOOLROOMS
2          CALL ASTLRM(NTLRMS,TLRMNM,NWNDWS,NATNTS,MU,LAMBDA,
    +         STRLCN,BLDCST,INVCST,ANWFSA,ATNWGS,YRDPRM,NWRKRS,
    +         WRKLCN,WRKWGS,WLKRT,NSRVAR,SRVNMS,SAPRIM,
    +         WQ,WS,PO,PN,RHO,LS,LQ,ATRLCN,DTRWRK,ADSATR,
    +         WDTLRM,OPMSR,QTRCST,TTRCST,TOTCST,CS,CW,
    +         NWRKSA,CNTRD,PTRLCN,NTIMES,MNUVAR,OPTIMES)
   OUTPUT SHIPYARD MODEL
  CALL OSMDAS(NTLRMS,TLRMNM,NWNDWS,NATNTS,MU,
    +         LAMBDA,BLDCST,INVCST,ANWFSA,NSRVAR,SRVNMS,WQ,
    +          WS,PO,PN,RHO,LS,LQ,ATRLCN,DTRWRK,ADSATR,
    +         WDTLRM,OPMSR,QTRCST,TTRCST,
    +         TOTCST,CS,CW,NWRKSA,MNUVAR)
   CHECK TO SEE IF OPTIMAL LOCATIONS ARE CHOSEN OR IF USER
    WANTS PROGRAM TO END
  NTIMES=NTIMES+1
  CALL ISEND(NTLRMS,ATRLCN,PTRLCN,QUITIT,MNUVAR)
  OPTIMES = OPTIMES + 1
IF (QUITIT.EQ.1) GOTO 12
IF (QUITIT.EQ.2) GOTO 2
PAUSE
END


CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC


 GET VALID INPUT
SUBROUTINE GVLNPT(NTLRMS,TLRMNM,NWNDWS,NATNTS,MU,LAMBDA,
```

```
     +       STRLCN,BLDCST INVCST,ANWFSA ATNWGS,YRDPRM ,NWRKRS,
     +       WRKLCN,WRKWGS  ,WLKRT,NSRVAR,SRVNMS,
     +       SAPRIM,MNUVAR OPTIMES)

  OUTPUT
 INTEGER NTLRMS
 CHARACTER*2O TLRMNM(10)
 INTEGER NWNDWS( 10),NATNTS(1O)
 REAL MU(10),LAMBDA(1O)
 REAL STRLCN(10,2)
 REAL BLDCST( 10),INVCST(1O)
 REAL ANWFSA( 10),ATNWGS(10)
 REAL YRDPRM(4)
 INTEGER NWRKRS
 REAL WRKLCN(1OO,2)
 REAL WRKWGS
 REAL WLKRT
 INTEGER NSRVAR
 CHARACTER*2O SRVNMS(20)
 REAL SAPRIM(20,4)
 REAL ATRLCN(10,2)
 INTEGER MNUVAR
 INTEGER OPTIMES

  GET TOOLROOM DATA

 CALL   GTRDAT(NTLRMS,TLRMNM,NWNDWS,NATNTS,MU,LAMBDA,
     +       STRLCN,BLDCST,INVCST,ATNWGS  ,MNUVAR,ATRLCN  ,OPTIMES)
  GET WORKER DATA
 IF (MNUVAR.GE.2) THEN
   CALL   GWKDAT(NWRKRS,WRKLCN,WRKWGS,WLKRT,MNUVAR,OPTIMES)
 END IF
  GET SERVICE AREA DATA
 IF (MNUVAR.GE.2) THEN
   CALL   GSADAT(NSRVAR,SRVNMS,SAPRIM,MNUVAR,0PTIMES)
 END IF
 IF (MNUVAR.GE.2) THEN
   DO 90 I=l,NTLRMS
     IF (OPTIMES.GT. 1.AND.ANWFSA( 1).NE.O.0) THEN
4       WRITE(*,35)TLRMNM( I),ANWFSA( I)
5              FORMAT(//,lX,'THE  AVERAGE',
     +        ' NUMBER OF WORKERS THAT USE 'A20,/,
     +        ' IS ',F8.0)
               WRITE(*,36)
6       FORMAT(/,lX,'ENTER A 1 TO CHANGE ELSE <RETURN> TO CONTINUE')
       READ(*,37,ERR=38) III
7              FORMAT(I2)
        GOTO 39
8         WRITE(*,91)
        GOTO 34
9       CONTINUE
          IF (III.NE.1) GOTO 90
     END IF
4            WRITE(*,45)TLRMNM(I)
5            FORMAT(//,lX,'INPUT  THE  AVERAGE',
     + '  NUMBER OF WORKERS THAT USE ',A20,/,lX,'REAL',
     +  /,lX,'ENTER:')
     READ(*,50,ERR=51  )ANWFSA(I)
0            FORMAT(F8.0)
        GOTO 55
```

```
1       WRITE(*,91)
        GOTO 44
5          WRITE(*,56)
6       FORMAT(/,lX,'ENTER A 1 TO CORRECT ELSE <RETURN> TO CONTINUE')
         READ(*,58,ERR=59) III
8              FORMAT(I2)
         (GOT0 60
9          WRITE(*,91)
         GOTO 55
0       CONTINUE
         IF  (III.EQ.1)  GOTO  44
0          CONTINUE
END IF
1       FORMAT(/,lX,'****ERROR****MISTYPED  DATA',/)
         END


cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc


   ANALYZE SHIPYARD TOOLROOMS
SUBROUTINE   ASTLRM(NTLRMS,TLRMNM,NWNDWS,NATNTS,MU,LAMEDA,
    +        STRLCN,BLDCST,INVCST,ANWFSA,  ATNWGS,YRDPRM  ,NWRKRS,
    +        WRKLCN,WRKWGS,WLKRT,NSRVAR,SRVNMS,SAPRIM,
    +        WQ,WS,PO,PN,RHO,LS,LQ,ATRLCN,DTRWRK  ,ADSATR,
    +    WDTLRM,OPMSR  ,QTRCST,TTRCST  ,TOTCST,CS,CW,
    +    NWRKSA,CNTRD  ,PTRLCN,NTIMES,MNUVAR,OPTIMES)

 INPUT
INTEGER NTLRMS
CHARACTER*2O  TLRMNM(10)
INTEGER NWNDWS( 10),NATNTS(1O)
REAL  MU(10),LAMBDA(1O)
REAL  STRLCN(10,2)
REAL  BLDCST( 10),INVCST(1O)
REAL  ANWFSA( 10),ATNWGS(10)
REAL  YRDPRM(4)
INTEGER NWRKRS
REAL  WRKLCN(1OO,2)
REAL WRKWGS
REAL WLKRT
INTEGER NSRVAR
CHARACTER*2O  SRVNMS(20)
REAL  SAPRIM(20,4)

 OUTPUT
        REAL  WQ(lO),WS( 1O),PO(10)
REAL  PN(5O,1O)
REAL RHO(lO),LS( 10),LQ(10).
REAL  ATRLCN(10,2)
REAL  DTRWRK(10,1OO)
REAL  ADSATR(2O,10)
REAL  WDTLRM(10)
REAL OPMSR(10)
REAL  QTRCST(10)
REAL  TTRCST(10)
REAL TOTCST
REAL CS(10),CW(1O)
INTEGER NWRKSA(20)
REAL  CNTRD(20,2)
```

```
REAL PTRLCN(10,2)
 OTHER INPUT
INTEGER NTIMES
INTEGER MNUVAR
INTEGER OPTIMES


 SET UP SHIPYARD MODEL
IF (MNUVAR.GE.2) THEN
  CALL SETMOD(NTLRMS,TLRMNM}STRLCN,YRDPRM,NWRKRS,WRKLCN,
    +       NSRVAR,SRVNMS,SAPRIM,ATRLCN,DTRWRK,ADSATR,  WDTLRM,
    +       OPMSR,QTRCST ,NWRKSA,CNTRD ,PTRLCN,
    +       NTIMES,MNUVAR  ,OPTIMES)
       END IF
 DO QUEUING THOERY
CALL DQTHRY(NTLRMS,NWNDWS,NATNTS,MU,LAMBDA,
    +       WQ,WS,PO,PN,RHO,LS,LQ)
 CALCULATE COST OF CURRENT MODEL
IF (MNUVAR.GE.2) THEN
  CALL CSTMOD(NTLRMS,LAMBDA,WQ,WS,BLDCST, INVCST,
    +         ANWFSA,ATNWGS,WRKWGS,WLKRT,NSRVAR,ADSATR,WDTLRM,
    +         OPMSR,GITRCST,TTRCST,TOTCST,
    +         CS,CW,MNUVAR)
       END IF
        END


cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc


   OUTPUT SHIPYARD MODEL
  SUBROUTINE OSMDAS(NTLRMS,TLRMNM,NWNDWS,NATNTS,MU,
    +        LAMBDA,BLDCST,INVCST,ANWFSA  ,NSRVAR,SRVNMS,WQ,
    +         WS,PO,PN,RHO ,LS,LQ,ATRLCN ,DTRWRK,ADSATR ,
    +        WDTLRM,OPMSR,QTRCST,TTRCST ,
    +        TOTCST,CS,CW,NWRKSA,MNUVAR )


INTEGER NTLRMS
CHARACTER*2O TLRMNM(10)
INTEGER NWNDWS( 10),NATNTS(1O)
REAL MU(10),LAMBDA(1O)
REAL BLDCST( 10),INVCST(1O)
REAL ANWFSA(10)
INTEGER NSRVAR
CHARACTER*2O SRVNMS(20)
REAL WQ(lO).,WS( 1O),PO(1O)
        REAL PN(5O,1O)
REAL RHO(lO),LS( 1O),LQ(1O)
REAL ATRLCN(10,2)
REAL DTRWRK(10,1OO)
REAL ADSATR(20,10)
REAL WDTLRM(10)
REAL OPMSR(10)
REAL QTRCST(10)
REAL TTRCST(10)
REAL TOTCST
REAL CS(l0),cw(lo)
INTEGER NWRKSA(20)
INTEGER MNUVAR
```

```
       INTEGER INCR(10)

CHARACTER GRAPH(30,60)

DO 500 I=1,NTLRMS
  IF (MNUVAR.GE.1) THEN
    WRITE(*,10)TLRMNM(I)
0            FORMAT(//,1X,'***STATISTICS FOR ',A20,)
    IF (MNUVAR.GE.3) THEN
      WRITE(*,15)ATRLCN(I,1),ATRLCN(I,2)
5            FORMAT(//,1X,'***OPTIMAL LOCATION IS (',F8.4,',',F8.4,')')
             END IF
    WRITE(*,20)MU(I)
0            FORMAT(//,80('*'),/,5X,'MEAN SERVICE RATE - ',F8.2)
    WRITE(*,30)LAMBDA(I)
0            FORMAT(5X,'MEAN ARRIVAL RATE - ',F8.2)
    WRITE(*,40)WQ(I)
0            FORMAT(5X,'MEAN TIME SPENT WAITING IN LINE - ',F8.2)
    WRITE(*,50)WS(I)
0            FORMAT(5X,'MEAN TIME SPENT WAITING IN SYSTEM - ',F8.2)
    WRITE(*,60)P0(I)
0            FORMAT(5X,'PROBABILITY OF EMPTY QUEUE - ',F8.2)
    WRITE(*,70)RHO(I)
0            FORMAT(5X,'UTILIZATION FACTOR - ',F8.2)
    WRITE(*,80)LS(I)
0            FORMAT(5X,'EXPECTED NUMBER OF UNITS IN SYSTEM - ',F8.2)
    WRITE(*,85)LQ(I)
5            FORMAT(5X,'MEAN LENGTH OF QUEUE - ',F8.2,/,80('*'))
    WRITE(*,90)
0        FORMAT(//,1X,'CHOOSE THE FORM TO DISPLAY P(N)')
    WRITE(*,95)
5        FORMAT(10X,'1 - TABLE')
    WRITE(*,100)
00       FORMAT(10X,'2 - CHART')
    READ(*,105)IOPT
05       FORMAT(I2)
         END IF

 PLOT OF PN
         WRITE(*,107)
07       FORMAT(/)
         SUM=0
         ZERO2=0
         JJ=0
         ZERO=0
        DO 106 M=1, 30
         DO 106 N=1,60
            GRAPH(M,N)=' '
06       CONTINUE
        DO 110 K=1,50
        SUM=SUM+PN(K,I)
        IF (SUM.GE.0.99)ZERO=ZERO+1.
        IF (SUM.GE.0.9999)ZERO2=ZERO2+1.
10     CONTINUE
         UNITS=50.-ZERO
         IUNIT=50.-ZERO2
        IF (UNITS.LE.5.)J=5
        IF (UNITS.LE.10.0.AND.UNITS.GT.5.0)J=10
        IF (UNITS.LE.20.0.AND.UNITS.GT.10.0)J=20
        IF (UNITS.LE.30.0.AND.UNITS.GT.20.0)J=30
```

```
 IF (UNITS.LE.30.0.AND.UNITS.GT.40.0)J=30
IF (IOPT.EQ.1) THEN
WRITE(*,115)
FORMAT(1X,'NUMBER OF UNITS',3X,'PROBABILITY OF OCCURRENCE',/)
DO 150 K=1,IUNIT
  WRITE(*,125)K,PN(K,I)
  FORMAT(8X,I2,10X,F12.4)
        CONTINUE
END IF
  ICHAR=INT(60./J)
IF (IOPT.EQ.2) THEN
 ICHAR=INT(60./J)
 DO 165 M=1,J
  DO 165 N=1,ICHAR
   JJ=JJ+1
   DO 165 K=1,30
     IF ((K*3.333).LE.(PN(M,I)*100)) THEN
       GRAPH(K,JJ)= '*'
     ELSE
       GRAPH(K,JJ)= ' '
     END IF
        CONTINUE
 WRITE(*,167)
  FORMAT(1X,'PLOT OF P(N) - PROBABILITY OF N UNITS IN SYSTEM',/)
 WRITE(*,168)
 FORMAT(/)
 N=5
 IYAXIS=120
 ICOUNT=J*ICHAR
 DO 190 L=30,1,-1
   IF (L.EQ.(6*N)) THEN
     N=N-1
     IYAXIS=IYAXIS-20
     WRITE(*,170)IYAXIS,(GRAPH(L,M),M=1,ICOUNT)
             FORMAT(16X,I3,T20,60A1)
   ELSE IF (L.NE.(6*N).AND.L.NE.23.AND.L.NE.22)THEN
     WRITE(*,175)(GRAPH(L,M),M=1,ICOUNT)
             FORMAT(T20,60A1)
   ELSE IF (L.EQ.23) THEN
        WRITE(*,178)(GRAPH(L,M),M=1,ICOUNT)
        FORMAT(1X,'PROBABILITY',T20,60A1)
   ELSE IF (L.EQ.22) THEN
        WRITE(*,179)(GRAPH(L,M),M=1,ICOUNT)
        FORMAT(2X,'(PERCENT)',T20,60A1)
   END IF

        CONTINUE
  DO 193 N=1,J
    INCR(N)=N
  CONTINUE
  IF (J.EQ.5)WRITE(*,194)(INCR(N),N=1,J)
  IF (J.EQ.10)WRITE(*,195)(INCR(N),N=1,J)
  IF (J.EQ.20)WRITE(*,195)(INCR(N),N=1,J,2)
  IF (J.EQ.30)WRITE(*,195)(INCR(N),N=1,J,3)
  FORMAT(T26,I2,10X,I2,10X,I2,10X,I2,10X,I2)
       FORMAT(T22,I2,4X,I2,4X,I2,4X,I2,4X,I2,4X,I2,4X,I2,4X,I2,4X,I2,
+4X,I2)
  WRITE(*,196)
  FORMAT(/,T48,'UNITS')
```

```
         END IF
   IF (MNUVAR.GE.2) THEN
     WRITE(*,255)OPMSR(I)
55       FORMAT(//,80('*'),/,1X,'OPTIMAL MEAN SERVICE RATE    - ',F12.4)
     WRITE(*,260)CW(I)
60       FORMAT(1X,'WAITING COST PER UNIT           - ',F15.2)
     WRITE(*,265)CS(I)
65       FORMAT(1X,'SERVICING COST PER UNIT         - ',F15.2)
     WRITE(*,270)QTRCST(I)
70       FORMAT(1X,'TOOLROOM COST DUE TO QUEUEING- ',F15.2)
     WRITE(*,280)TTRCST(I)
80       FORMAT(1X,'TOOLROOM COST TOTAL            - ',F15.2,
     +/,80('*'),//)
      END IF
00       CONTINUE
IF (MNUVAR.GE.2) THEN
  WRITE(*,510)TOTCST
10       FORMAT(1X,'COST FOR ALL TOOLROOMS          -',F15.2,
     +/,1X,80('*'),//)
END IF
         END


CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC


   CHECK TO SEE IF OPTIMAL LOCATIONS ARE CHOSEN OR IF USER
    WANTS PROGRAM TO END
SUBROUTINE ISEND(NTLRMS,ATRLCN,PTRLCN,QUITIT,MNUVAR)
REAL ATRLCN(10,2)
REAL PTRLCN(10,2)
INTEGER QUITIT
INTEGER MNUVAR
INTEGER CNTR
CNTR=0
 IF ACTUAL LOCATION = PERFECT LOCATION THEN
IF (MNUVAR.GE.3) THEN
 DO 10 I=1,NTLRMS
  IF ((ABS(ATRLCN(I,1)-PTRLCN(I,1)).LT.0.01).AND.
    +        (ABS(ATRLCN(I,2)-PTRLCN(I,2)).LT.0.01))  THEN
         CNTR = CNTR + 1
  END IF
0         CONTINUE
 IF (CNTR.EQ.NTLRMS) THEN
 TELL USER ALL LOCATIONS ARE OPTIMAL
  WRITE(*,15)
5          FORMAT(1X,'ALL TOOLROOM LOCATIONS ARE OPTIMAL')
         END IF
END IF
 ASK IF USER WANTS TO QUIT SESSION
9        WRITE(*,20)
0        FORMAT(//,1X,'ENTER ONE OF THE FOLLOWING TO:')
WRITE(*,21)
1        FORMAT(10X,'1 - CONTINUE WITH MODEL')
WRITE(*,22)
2        FORMAT(10X,'2 - RETURN TO STARTUP MENU')
WRITE(*,23)
3        FORMAT(10X,'3 - QUIT')
WRITE(*,24)
4        FORMAT(1X,'INTEGER',/,1X,'ENTER:',/)
```

```
READ(*,25,ERR=26)QUITIT
5        FORMAT(I4)
        GOTO 27
6       WRITE(*,28)
        GOTO 19
7       CONTINUE
8       FORMAT(,/,1X,'***ERROR***MISTYPED DATA',/)
END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC


 GET TOOLROOM DATA
SUBROUTINE GTRDAT(NTLRMS,TLRMNM,NWNDWS,NATNTS,MU,LAMBDA,
    +      STRLCN,BLDCST,INVCST,ATNWGS,MNUVAR,ATRLCN,OPTIMES)

 OUTPUT
INTEGER NTLRMS
CHARACTER*20 TLRMNM(10)
INTEGER NWNDWS(10),NATNTS(10)
REAL MU(10),LAMBDA(10)
REAL STRLCN(10,2)
REAL BLDCST(10),INVCST(10)
REAL ATNWGS(10)
REAL ATRLCN(10,2)
INTEGER MNUVAR
INTEGER OPTIMES

 GET #TOOLROOMS
IF (OPTIMES.GT.1) THEN
           WRITE(*,2)NTLRMS
           FORMAT(//,1X,'CURRENT NUMBER OF TOOLROOMS IS',I3)
           WRITE(*,3)
        FORMAT(/,1X,'ENTER A 1 TO CHANGE ELSE <RETURN> TO CONTINUE')
  READ(*,4,ERR=5)III
           FORMAT(I2)
        GOTO 6
        WRITE(*,1000)
        GOTO 1
        CONTINUE
           IF (III.NE.1) GOTO 17
END IF

         WRITE(*,8)
         FORMAT(//,1X,'INPUT NUMBER OF TOOLROOMS ',/,1X
     +,'INTEGER',/,1X,'ENTER:')
READ(*,9,ERR=10)NTLRMS
         FORMAT(I4)
        GOTO 11
0       WRITE(*,1000)
        GOTO 7
1       CONTINUE
2        WRITE(*,13)
3       FORMAT(/,1X,'ENTER 1 TO CORRECT ELSE <RETURN> TO CONTINUE')
        READ(*,14,ERR=15)III
4        FORMAT(I2)
        GOTO 16
5       WRITE(*,1000)
        GOTO 12
6       CONTINUE
```

```
7       DO 999 I=1, NTLRMS
 GET TOOLROOM NAMES
    IF (OPTIMES.GT.1) THEN
      WRITE(*, 100)TLRMNM(I)
00           FORMAT(//,lX,'TOOLROOM NAME IS' ,A20)
35            WRITE(*,136)
36     FORMAT(/,lX,'ENTER 1 TO CHANGE ELSE <RETURN> TO CONTINUE')
      READ(*s 138,ERR=139)III
38               FORMAT(I2)
       GOTO 140
39      WRITE(*,1000)
        GOTO 135
40      CONTINUE
        IF (111.NE.1) GOTO 200
    END IF
50               WRITE(*,175)I
75               FORMAT(//,lX,'INPUT NAME OF TOOLROOM # ',12,':',/,lX,'CHARACTE

    +/,1X,'ENTER:')
        READ(*, 180,ERR=181 )TLRMNM( I)
80               FORMAT(A20)
       GOTO 182
81     WRITE(*,1000)
       GOTO 150
02     CONTINUE
87     WRITE(*,189)
89     FORMAT(/,lX,'ENTER 1 TO CORRECT ELSE <RETURN> TO CONTINUE')
         READ(*,  190,ERR=191)II  I
90             FORMAT(I2)
       GOTO 192
91     WRITE(*,1000)
       GOTO 187
92     CONTINUE
      IF  III.EQ.1) GOTO 150
00               WRITE(*,205)TLRMNM(I)
05      FORMAT(//,lX,'INPUT THE FOLLOWING DATA FOR', A20)
 GET #WINDOWS
    IF (OPTIMES.GT.1) THEN
        wRITE(*,330)NWNDWS(I)
30               FORMAT(//,5X,'NUMBER OF WINDOWS IS',13)
35             WRITE(*,336)
36     FORMAT(//,lX,'ENTER 1 TO CHANGE ELSE <RETURN> TO CONTINUE')
         READ(*,338,ERR=339)III
       GO TO 340
39     WRITE(*,1000)
       GOTO 335
40     CONTINUE
30             FORMAT(I2)
      IF (III.NE.1) GOTO 499
    END IF
99      WRITE(*,400)
00               FORMAT(//,5X,'NUMEER  OF  WINDOWS',/,lX,'INTEGER',
    +/,lX,'ENTER:')
        READ(*,401,ERR=402)NWNDWS(I)
01               FORMAT(I4)
       GOTO 455
02      WRITE(*,1000)
       GOTO 399
55             WRITE(*,456)
56       FORMAT(/,lX,'ENTER 1 TO CORRECT ELSE <RETURN> TO CONTINUE')
```

```
      READ(*,458,ERR=459)III
58            FORMAT(I2)
       GOTO 460
59      WRITE(*,1000)
        GOTO 455
60     CONTINUE
      IF (III.EQ.1) GOTO 399
 GET #ATTENDANTS
99          IF (OPTIMES.GT.1) THEN
             WRITE(*,500)NATNTS(I)
00            FORMAT(//,5X,'NUMBER OF ATTENDANTS IS ',I3)
      IF (MNUVAR.GE.2.AND.ATNWGS(I).NE.0.0) THEN
              WRITE(*,522)ATNWGS(I)
22            FORMAT(/,5X,'ATTENDANTS HOURLY WAGE IS ',F8.2)
      END IF
35            WRITE(*,536)
36     FORMAT(/,1X,'ENTER 1 TO CHANGE ELSE <RETURN> TO CONTINUE')
      READ(*,538,ERR=539)III
38            FORMAT(I2)
       GOTO 540
39     WRITE(*,1000)
       GOTO 535
40     CONTINUE
      IF (III.NE.1) GOTO 599
    END IF
44      WRITE(*,550)
50              FORMAT(//,5X,'NUMBER OF ATTENDANTS ',/,1X,'INTEGER',
    +/,1X,'ENTER:')
        READ(*,551,ERR=552)NATNTS(I)
51              FORMAT(I4)
       GOTO 553
52      WRITE(*,1000)
       GOTO 544
53      CONTINUE
 GET ATTENDANTS WAGES
        IF (MNUVAR.GE.2) THEN
54                WRITE(*,555)
55                FORMAT(//,5X,'ATTENDANTS HOURLY WAGE ',/,1X,'REAL',
    +/,1X,'ENTER:')
          READ(*,556,ERR=557)ATNWGS(I)
56                FORMAT(F8.0)
       GOTO 558
57     WRITE(*,1000)
       GOTO 554
58      CONTINUE
        END IF
59            WRITE(*,560)
60     FORMAT(/,1X,'ENTER 1 TO CORRECT ELSE <RETURN> TO CONTINUE')
      READ(*,561,ERR=562)III
61            FORMAT(I2)
       GOTO 563
62     WRITE(*,1000)
       GOTO 559
63        CONTINUE
      IF (III.EQ.1) GOTO 544
99          IF (OPTIMES.GT.1) THEN
              WRITE(*,600)MU(I)
00            FORMAT(//,5X,'MEAN SERVICE RATE IS',F8.4)
35            WRITE(*,636)
36     FORMAT(/,1X,'ENTER 1 TO CHANGE ELSE <RETURN> TO CONTINUE')
```

```
      READ(*,638,ERR=639)III
38           FORMAT(I2)
      GOTO 640
39    WRITE(*,1000)
      GOTO 635
40    CONTINUE
     IF (III.NE.1) GOTO 699
   END IF
 GET MU
44           WRITE(*,645)
45           FORMAT(//,5X,'MEAN SERVICE RATE ',/,1X,'REAL',
   +/,1X,'ENTER:')
     READ(*,646,ERR=647)MU(I)
46           FORMAT(F8.0)
      GOTO 650
47    WRITE(*,1000)
      GOTO 644
50           WRITE(*,656)
56    FORMAT(/,1X,'ENTER 1 TO CORRECT ELSE! <RETURN> TO CONTINUE')
     READ(*,658,ERR=659)III
58           FORMAT(I2)
      GOTO 660
59    WRITE(*,1000)
      GOTO 650
60    CONTINUE
     IF (III.EQ.1) GOTO 644
 GET LAMBDA
99          IF (OPTIMES.GT.1) THEN
               WRITE(*,700)LAMBDA(I)
00             FORMAT(//,5X,'MEAN ARRIVAL RATE IS ',F8.4)
35            WRITE(*,736)
36    FORMAT(/,1X,'ENTER 1 TO CHANGE ELSE <RETURN> TO CONTINUE')
     READ(*,738,ERR=739)III
38            FORMAT(I2)
      GOTO 740
39     WRITE(*,1000)
      GOTO 735
40    CONTINUE
     IF (III.NE.1) GOTO 799
   END IF
44             WRITE(*,750)
50             FORMAT(//,5X,'MEAN ARRIVAL RATE ',/,1X,'REAL',
   +/,1X,'ENTER:')
       READ(*,751,ERR=752)LAMBDA(I)
51             FORMAT(F8.0)
      GOTO 753
52    WRITE(*,1000)
      GOTO 744
53    WRITE(*,756)
56    FORMAT(/,1X,'ENTER 1 TO CORRECT ELSE <RETURN> TO CONTINUE')
     READ(*,758,ERR=759)III
58            FORMAT(I2)
      GOTO 760
59    WRITE(*,1000)
      GOTO 753
60    CONTINUE
      IF (III.EQ.1) GOTO 744
 GET BUILDING COST
99          IF (MNUVAR.GE.2) THEN
    IF (OPTIMES.GT.1.AND.BLDCST(I).NE.0.0) THEN
```

```
                 WRITE(*,800)BLDCST(I)
00               FORMAT(//,5X,'BUILDING COST IS ',F14.2)
35               WRITE(*,836)
36        FORMAT(/,1X,'ENTER 1 TO CHANGE ELSE <RETURN> TO CONTINUE')
         READ(*,838,ERR=839)III
38               FORMAT(I2)
          GO TO 840
39        WRITE(*,1000)
          GOTO 835
40        CONTINUE
         IF (III.NE.1) GOTO 899
       END IF
44             WRITE(*,850)
50                     FORMAT(//,5X,'BUILDING COST ',/,1X,'REAL',/,1X,'ENTER:')
             READ(*,851,ERR=852)BLDCST(I)
51                     FORMAT(F8.0)
          GOTO 853
52        WRITE(*,1000)
          GOTO 844
53         WRITE(*,856)
56        FORMAT(/,1X,'ENTER 1 TO CORRECT ELSE <RETURN> TO CONTINUE')
         READ(*,858,ERR=859)III
58               FORMAT(I2)
          GOTO 860
59        WRITE(*,1000)
          GOTO 853
60        CONTINUE
         IF (III.EQ.1) GOTO 844
 GET INVENTORY COST
99             IF (OPTIMES.GT.1.AND.INVCST(I).NE.0.0) THEN
                 WRITE(*,900)INVCST(I)
00               FORMAT(//,5X,'INVENTORY COST IS ',F14.2)
35               WRITE(*,936)
36        FORMAT(/,1X,'ENTER 1 TO CHANGE ELSE <RETURN> TO CONTINUE')
         READ(*,938,ERR=939)III
38               FORMAT(I2)
          GOTO 940
39        WRITE(*,1000)
          GOTO 935
40        CONTINUE
         IF (III.NE.1) GOTO 999
       END IF
44                     WRITE(*,950)
50                     FORMAT(//,5X,'INVENTORY COST ',/,1X,'REAL',/,1X,'ENTER:')
             READ(*,951,ERR=952)INVCST(I)
51                     FORMAT(F8.0)
          GOTO 955
52        WRITE(*,1000)
          GOTO 944
55               WRITE(*,956)
56        FORMAT(/,1X,'ENTER 1 TO CORRECT ELSE <RETURN> TO CONTINUE')
         READ(*,958,ERR=959)III
58               FORMAT(I2)
          GOTO 960
59        WRITE(*,1000)
          GOTO 955
60        CONTINUE
         IF (III.EQ.1) GOTO 944
 END IF
99         CONTINUE
```

```
000    FORMAT(/,1X,'****ERROR****MISTYPED DATA',/)
       END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC


 GET WORKER DATA
SUBROUTINE GWKDAT(NWRKRS,WRKLCN,WRKWGS,
    +                        WLKRT,MNUVAR,OPTIMES)
 OUTPUT
INTEGER NWRKRS
REAL WRKLCN(100,2)
REAL WRKWGS
REAL WLKRT
INTEGER MNUVAR
INTEGER OPTIMES
 GET #WORKERS
IF (MNUVAR.GE.3) THEN
    IF (OPTIMES.GT.1.AND.NWRKRS.NE.0) THEN
         WRITE(*,10)NWRKRS
0        FORMAT(//,1X,'NUMBER OF WORKERS IS ',I4)
5              WRITE(*,36)
6       FORMAT(/,1X,'ENTER 1 TO CHANGE ELSE <RETURN> TO CONTINUE')
       READ(*,38,ERR=39)III
8              FORMAT(I2)
        GOTO 40
9       WRITE(*,900)
        GOTO 35
0       CONTINUE
       IF (III.NE.1) GOTO 599
    END IF
44       WRITE(*,510)
10       FORMAT(//,1X,'INPUT NUMBER OF WORKERS ',/,1X,'INTEGER',
    +/,1X'ENTER:')
  READ(*,511,ERR=512)NWRKRS
11       FORMAT(I4)
        GOTO 555
12      WRITE(*,900)
        GOTO 444
55             WRITE(*,556)
56      FORMAT(/,1X,'ENTER 1 TO CORRECT ELSE <RETURN> TO CONTINUE')
        READ(*,558,ERR=559)III
58             FORMAT(I2)
        GOTO 560
59      WRITE(*,900)
        GOTO 555
60      CONTINUE
       IF (III.EQ.1) GOTO 444
END IF
99       IF (MNUVAR.GE.4) THEN
  DO 699 I=1,NWRKRS
 GET WORKER LOCATIONS
        WRITE(*,630)I
30              FORMAT(1X,'INPUT FOLLOWING FOR WORKER #',I4)
        WRITE(*,635)
35              FORMAT(5X,'LOCATION')
 GET SERVICE AREA PERIMETERS
        WRITE(*,640)
40              FORMAT(7X,'X-COORD.')
```

```
        READ(*,645)WRKLCN(I,1)
45              FORMAT(F8.0)
        WRITE(*,650)
50              FORMAT(7X,'Y-COORD. ')
        READ(*,655)WRKLCN(I,2)
55              FORMAT(F8.0)
99          CONTINUE
END IF
 GET WORKERS WAGES
IF (MNUVAR.GE.2) THEN
    IF (OPTIMES.GT.1.AND.WRKWGS.NE.0.0) THEN
    .       WRITE(*,700)WRKWGS
00          FORMAT(//,5X,'WORKERS AVERAGE HOURLY WAGE IS',F8.2)
35              WRITE(*,736)
36        FORMAT(/,1X,'ENTER 1 TO CHANGE ELSE <RETURN> TO CONTINUE')
       READ(*,738,ERR=739)III
38              FORMAT(I2)
        GOTO 740
39      WRITE(*,900)
        GOTO 735
40      CONTINUE
       IF (III.NE.1) GOTO 799
    END IF
44          WRITE(*,745)
45          FORMAT(//,5X,'WORKERS AVERAGE HOURLY WAGE ',/,1X,'REAL',
    +/,1X,'ENTER:')
  READ(*,746,ERR=747)WRKWGS
46          FORMAT(F8.0)
        GOTO 755
47      WRITE(*,900)
        GOTO 744
55       WRITE(*,756)
56      FORMAT(/,1X,'ENTER 1 TO CORRECT ELSE <RETURN> TO CONTINUE')
       READ(*,758,ERR=759)III
58              FORMAT(I2)
        GOTO 760
59      WRITE(*,900)
        GOTO 755
60      CONTINUE
       IF (III.EQ.1) GOTO 744
 GET WALK RATE IN FEET/MIN
99            IF (OPTIMES.GT.1.AND.WLKRT.NE.0.0) THEN
            WRITE(*,800)WLKRT
00          FORMAT(//,5X,'AVERAGE WALK RATE OF WORKERS IS',F8.2)
35              WRITE(*,836)
36      FORMAT(/,1X,'ENTER 1 TO CHANGE ELSE <RETURN> TO CONTINUE')
       READ(*,838,ERR=839)III
38              FORMAT(I2)
        GOTO 840
39      WRITE(*,900)
        GOTO 835
40      CONTINUE
        IF (III.NE.1) GOTO 899
    END IF
44          WRITE(*,845)
45          FORMAT(//,5X,'AVERAGE WALK RATE OF WORKERS ',/,1X,'REAL',
    +/,1X,'ENTER:')
  READ(*,846,ERR=847)WLKRT
46          FORMAT(F8.0)
        GOTO 855
```

```
47      WRITE(*,900)
        GOTO 844
55             WRITE(*,856)
56      FORMAT(/,1X,'ENTER 1 TO CORRECT ELSE <RETURN> TO CONTINUE')
       READ(*,858,ERR=859)III
58             FORMAT(I2)
        GOTO 860
59      WRITE(*,900)
        GOTO 855
60      CONTINUE
       IF (III.EQ.1) GOTO 844
END IF
99       CONTINUE
00      FORMAT(/,1X,'***ERROR***MISTYPED DATA',/)
END


CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC


 GET SERVICE AREA DATA
SUBROUTINE GSADAT(NSRVAR,SRVNMS,SAPRIM,MNUVAR,OPTIMES)

 INPUT OUTPUT
INTEGER NSRVAR
CHARACTER*20 SRVNMS(20)
REAL SAPRIM(20,4)
INTEGER MNUVAR
INTEGER OPTIMES
 GET #SERVICE AREAS
IF (MNUVAR.GE.2) THEN
    IF (OPTIMES.GT.1.AND.NSRVAR.NE.0) THEN
          WRITE(*,30)NSRVAR
0         FORMAT(//,1X,'NUMBER OF SERVICE AREAS IS ',I3)
5              WRITE(*,36)
6       FORMAT(/,1X,'ENTER 1 TO CHANGE ELSE <RETURN> TO CONTINUE')
       READ(*,38,ERR=39)III
8              FORMAT(I2)
        GOTO 40
9       WRITE(*,601)
        GOTO 35
0       CONTINUE
       IF (III.NE.1) GOTO 99
   END IF
4         WRITE(*,50)
0         FORMAT(//,1X,'INPUT NUMBER OF SERVICE AREAS ',/,1X,'INTEGER',
   +/,1X,'ENTER:')
  READ(*,51,ERR=52)NSRVAR
1         FORMAT(I4)
        GOTO 55
2       WRITE(8,601)
        GOTO 44
5              WRITE(*,56)
6       FORMAT(/,1X,'ENTER 1 TO CORRECT ELSE <RETURN> TO CONTINUE')
       READ(*,58,ERR=59)III
8              FORMAT(I2)
        GOTO 60
9       WRITE(*,601)
        GOTO 55
0       CONTINUE
```

```
      IF (III.EQ.1) GOTO 44
END IF
9        IF (MNUV9R.GE.2) THEN
  DO 600 I=l,NSRVAR
 GET SERVICE AREA NAMES
    IF (OPTIMES.GT. 1.AND.SRVNMS(1).NE.' ') THEN
                  WRITE(*, 1OO)I,SRVNMS(I)
00              FORMAT(//,lX,.'NAME OF SERVICE AREA # ',13,
    +             ' IS ',A20)
35            WRITE(*,136)
36     FORMAT(/,lX,'ENTER 1 TO CHANGE ELSE <RETURN> TO CONTINUE')
      READ(*, 138,ERR=139)III
38            FORMAT(12)
       GOTO 140
39     WRITE(*,601)
       GOTO 135
40     CONTINUE
      IF (III.NE.1) GOTO 199
    END IF
44                WRITE(*,150)I
50              FORMAT(//,lX,'INPUT NAME OF SERVICE AREA # ',12,':',/,lX,
    +   'CHARACTER',/,lX,'ENTER: ')
          READ(*,  151,ERR=152)SRVNMS(1)
51                FORMAT(A20)
       GOTO 155
52     WRITE(*,601)
       GOTO 144
55      WRITE(*,156)
56     FORMAT(/,lX,'ENTER 1 TO CORRECT ELSE <RETURN> TO CONTINUE')
      READ(*, 158,ERR=159)III
58            FORMAT(I2)
       GOTO 160
59     WRITE(*,601)
       GOTO 155
60     CONTINUE
      IF (III.EQ.1) GOTO 144
99              IF (MNUVAR.GE.3) THEN
          WRITE(*,200)SRVNMS(I)
00              FORMAT(//,lX,'INPUT THE FOLLOWING DATA FOR SERVICE AREA: ',A20)
 GET SERVICE AREA PERIMETERS
    IF (OPTIMES.GT. 1.AND.SAPRIM( 1,1).NE.O.0) THEN
                  WRITE(*,201)SAPRIM(I,1)
01      FORMAT(5X, 'X-COORD. OF LEFTMOST BOUNDARY IS ',F8.4)
          WRITE(*,202)SAPRIM(  192)
02      FORMAT(5X, 'Y-COORD. OF BOTTOMMOST BOUNDARY IS ',FE.4)
          WRITE(*,203)SAPRIM(  I,3)
03      FORMAT(5X, 'X-COORD. OF RIGHTMOST BOUNDARY IS ',F8.4)
          WRITE(*,204)SAPRIM(  I,4)
04      FORMAT(SX, 'Y-COORD. OF TOPMOST BOUNDARY IS ',FS.4)'
35            WRITE(*,236)
36     FORMAT(/,lX,'ENTER 1 TO CHANGE ELSE <RETURN> TO CONTINUE')
      READ(*,238,ERR=239 )III
38            FORMAT(I2)
       GOTO 240
39     WRITE(*,602)
       GOTO 235
40     CONTINUE
      IF (III.NE.1) GOTO 600
    END IF
44                WRITE(*,245)
```

```
45       FORMAT(//,5X,'X-COORD. OF LEFTMOST BOUNDARY ',/,1X,'REAL',
    +/,1X,'ENTER:')
           READ(*,246,ERR=247)SAPRIM(I,1)
46                  FORMAT(F8.0)
        GO TO 249
47      WRITE(*,601)
        GOTO 244
49         WRITE(*,250)
50       FORMAT(//,5X,'Y-COORD. OF BOTTOMMOST BOUNDARY ',/,1X,'REAL',
    +/,1X,'ENTER:')
           READ(*,255,ERR=256)SAPRIM(I,2)
55                  FORMAT(F8.0)
        GOTO 259
56      WRITE(*,601)
        GOTO 249
59         WRITE(*,260)
60       FORMAT(//,5X,'X-COORD. OF RIGHTMOST BOUNDARY ',/,1X,'REAL',
    +/,1X,'ENTER:')
           READ(*,265,ERR=266)SAPRIM(I,3)
65                  FORMAT(F8.0)
        GOTO 269
66      WRITE(*,601)
        GOTO 259
69         WRITE(*,270)
70       FORMAT(//,5X,'Y-COORD. OF TOPMOST BOUNDARY ',/,1X,'REAL',
    +/,1X,'ENTER:')
           READ(*,271,ERR=272)SAPRIM(I,4)
71                  FORMAT(F8.0)
        GOTO 275
72      WRITE(*,601)
        GOTO 269
75       WRITE(*,276)
76      FORMAT(/,1X,'ENTER 1 TO CORRECT ELSE <RETURN> TO CONTINUE')
       READ(*,278,ERR=279)III
78              FORMAT(I2)
        GOTO 280
79      WRITE(*,601)
        GOTO 275
80      CONTINUE
       IF (III.EQ.1) GOTO 244
         END IF
00         CONTINUE
END IF
01      FORMAT(/,1X,'***ERROR***MISTYPED DATA',/)
END


CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC


 SET UP SHIPYARD MODEL
SUBROUTINE SETMOD(NTLRMS,TLRMNM,STRLCN,YRDPRM,NWRKRS,WRKLCN,
    +         NSRVAR,SRVNMS,SAPRIM,ATRLCN,DTRWRK,ADSATR,WDTLRM,OPMSR,
    +         QTRCST,NWRKSA,CNTRD,PTRLCN,NTIMES,MNUVAR,OPTIMES)
 INPUT
INTEGER NTLRMS
CHARACTER*20 TLRMNM(10)
REAL STRLCN(10,2)
REAL YRDPRM(4)
INTEGER NWRKRS
```

```
REAL WRKLCN(100,2)
INTEGER NSRVAR
CHARACTER*20 SRVNMS(20)
REAL SAPRIM(20,4)
 OUTPUT
REAL ATRLCN(10,2)
REAL DTRWRK(10,100)
REAL ADSATR(20,10)
REAL WDTLRM(10)
REAL OPMSR(10)
REAL QTRCST(10)
INTEGER NWRKSA(20)
REAL CNTRD(20,2)
REAL PTRLCN(10,2)
 OTHER INPUT
INTEGER NTIMES
INTEGER MNUVAR
INTEGER OPTIMES
 FIND #WORKERS IN AND CENTROID IN EACH SERVICE AREA
IF (MNUVAR.GE.3) THEN
  CALL FNWCNT(NWRKRS,WRKLCN,NSRVAR,SRVNMS,SAPRIM,
    +          NWRKSA,CNTRD,MNUVAR,OPTIMES)
 CHOOSE TOOLROOM LOCATIONS
  CALL CTRLCN(NTLRMS,STRLCN,NSRVAR,ATRLCN,OPMSR,
    +       QTRCST,NWRKSA,CNTRD,PTRLCN,NTIMES,MNUVAR)
 CREATE DISTANCE MATRIX
  IF (MNUVAR.EQ.4) THEN
    CALL CREDIS(NTLRMS,NWRKRS,WRKLCN,ATRLCN,
    +       DTRWRK)
         END IF
      END IF
 CALCULATE AVERAGE DISTANCE FROM SERVICE AREA TO TOOLROOM
IF (MNUVAR.GE.2) THEN
CALL CAVDIS(NTLRMS,TLRMNM,NWRKRS,WRKLCN,NSRVAR,
    +     SRVNMS,SAPRIM,ATRLCN,DTRWRK,ADSATR,
    +     NWRKSA,CNTRD,NTIMES,MNUVAR,OPTIMES)
      END IF
 CALCULATE WEIGHTED AVERAGE DISTANCE TRAVELED TO TOOLROOM
      IF (MNUVAR.GE.3) THEN
  CALL WAVDIS(NTLRMS,NWRKRS,NSRVAR,
    +       ADSATR,WDTLRM,NWRKSA)
       END IF

      END


CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC


 DO QUEUING THOERY
SUBROUTINE DQTHRY(NTLRMS,NWNDWS,NATNTS,MU,LAMBDA,
    +        WQ,WS,PO,PN,RHO,LS,LQ)

 INPUT
INTEGER NTLRMS
INTEGER NWNDWS(10)
INTEGER NATNTS(10)
REAL MU(10)
REAL LAMBDA(10)
 OUTPUT
```

```
        REAL WQ(10),WS(10),PO(10)
REAL PN(50,10)
REAL RHO(10),LS(10),LQ(10)

INTEGER S(10)


DO 10 I=1,NTLRMS
  IF ((NATNTS(I).GT.1).OR.(NWNDWS(I).GT.1)) THEN
    CALCULATE S
        CALL CALCS(I,NWNDWS,NATNTS,S)
    DO MULTICHANNEL SINGLE PHASE QUEUING THEORY
        CALL MCHQTH(I,S,MU,LAMBDA,
    +                   PO,PN,RHO,LS,LQ,WS,WQ)
  ELSE
    DO SINGLE CHANNEL SINGLE PHASE QUEUING THEORY
        CALL SCHQTH(I,MU,LAMBDA,
    +                   WQ,WS,PO,PN,RHO,LS,LQ)
  END IF
0       CONTINUE
END


CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC


 CALCULATE COST OF CURRENT MODEL
SUBROUTINE CSTMOD(NTLRMS,LAMBDA,WQ,WS,BLDCST,INVCST,
    +         ANWFSA,ATNWGS,WRKWGS,WLKRT,NSRVAR,ADSATR,WDTLRM,
    +         OPMSR,QTRCST,TTRCST,TOTCST,
    +         CS,CW,MNUVAR)

INTEGER NTLRMS
REAL LAMBDA(10)
REAL WQ(10),WS(10)
REAL BLDCST(10),INVCST(10)
REAL ANWFSA(10),ATNWGS(10)
REAL WRKWGS
REAL WLKRT
INTEGER NSRVAR
REAL ADSATR(20,10)
REAL WDTLRM(10)
REAL OPMSR(10)
REAL QTRCST(10)
REAL TTRCST(10)
REAL TOTCST
REAL CS(10),CW(10)
INTEGER MNUVAR
 CALCULATE CS
CALL CALCCS(NTLRMS,WQ,WS,ATNWGS,WRKWGS,CS)
 CALCULATE CW
CALL CALCCW(NTLRMS,WQ,WRKWGS,CW)
 CALCULATE OPTIMUM MEAN SERVICE RATE FOR TOOLROOM
CALL COPMSR(NTLRMS,LAMBDA,OPMSR,CS,CW)
 CALCULATE QUEUING TOOLROOM COSTS
CALL CQTRCS(NTLRMS,ANWFSA,WRKWGS,WLKRT,NSRVAR,ADSATR,
    +            WDTLRM,QTRCST,CS,CW)
 CALCULATE TOTAL TOOLROOM COST
CALL CTTRCS(NTLRMS,BLDCST,INVCST,QTRCST,
    +            TTRCST)
```

```
 SUM TOTAL TOOLROOM COSTS
CALL CSTALL(NTLRMS,TTRCST,TOTCST)
       END


CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC


 FIND #WORKERS IN AND CENTROID IN EACH SERVICE AREA
SUBROUTINE FNWCNT(NWRKRS,WRKLCN,NSRVAR,SRVNMS,SAPRIM,
    +                            NWRKSA,CNTRD,MNUVAR,OPTIMES)
 INPUT
INTEGER NWRKRS
REAL WRKLCN(100,2)
INTEGER NSRVAR
CHARACTER*20 SRVNMS(20)
REAL SAPRIM(20,4)
 OUTPUT
INTEGER NWRKSA(20)
REAL CNTRD(20,2)
INTEGER MNUVAR
INTEGER OPTIMES
 INTERNAL VARIABLES
 SUM OF WORKERS THAT ARE IN SERVICE AREAS
INTEGER SWRKSA
 SUMS OF WORKERS XCOORDS AND YCOORDS
REAL XSWRK(10),YSWRK(10)
 THE PRINCIPLE USED TO FIND THE CENTROIDS WORKS
  ON THE X AND Y DIRECTIONS SEPARATELY
  IT ALSO TAKES MULTIPLE WORKERS AT THE SAME LOCATION

 ZERO OUT VARIABLES

DO 10 I=1,NSRVAR
  XSWRK(I)=0.
  YSWRK(I)=0.
  NWRKSA(I)=0
0       CONTINUE

 RECORD #WORKERS IN SERVICE AREAS AND SUM X AND Y LOCATION VALUES

IF (MNUVAR.EQ.3) THEN
         DO 100 I=1,NSRVAR
    IF (OPTIMES.GT.1.AND.NWRKSA(I).NE.0.0) THEN
               WRITE(*,30)SRVNMS(I),NWRKSA(I)
0      FORMAT(//,1X,'NUMBER OF WORKERS IN SERVICE AREA ',A20,/,
    +               ' IS ',I4)
5             WRITE(*,36)
6      FORMAT(/,1X,'ENTER 1 TO CHANGE ELSE <RETURN> TO CONTINUE')
      READ(*,38,ERR=39)III
8             FORMAT(I2)
       GOTO 40
9      WRITE(*,401)
       GOTO 35
0      CONTINUE
      IF (III.NE.1) GOTO 100
    END IF
4              WRITE(*,50)SRVNMS(I)
0     FORMAT(//,1X,'INPUT THE NUMBER OF WORKERS IN SERVICE AREA ',
    +   A20,/,1X,'INTEGER',/,1X,'ENTER:')
```

```
         READ(*,51,ERR=52)NWRKSA(I)
 1       FORMAT(I4)
        GO TO 55
 2      WRITE(*,401)
        GOTO 44
 5      WRITE(*,56)
 6      FORMAT(/,1X,'ENTER 1 TO CORRECT ELSE <RETURN> TO CONTINUE')
       READ(*,57,ERR=58)III
 7             FORMAT(I2)
        GO TO 59
 8      WRITE(*,401)
        GOTO 55
 9      CONTINUE
       IF (III.EQ.1) GOTO 44
 00        CONTINUE
 END IF

  CALCULATE CENTROIDS OF SERVICE AREAS

 IF (MNUVAR.EQ.3) THEN
  DO 350 I=1,NSRVAR
   CNTRD(I,1)=(SAPRIM(I,3)-SAPRIM(I,1))/2 + SAPRIM(I,1)
   CNTRD(I,2)=(SAPRIM(I,4)-SAPRIM(I,2))/2 + SAPRIM(I,2)
 50        CONTINUE
 END IF
 00       CONTINUE
 END IF
 01      FORMAT(,/,1X,'***ERROR***MISTYPED DATA',/)
 END


 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC


  CHOOSE TOOLROOM LOCATIONS
 SUBROUTINE CTRLCN(NTLRMS,STRLCN,NSRVAR,ATRLCN,OPMSR,
     +         QTRCST,NWRKSA,CNTRD,PTRLCN,NTIMES,MNUVAR)

 INTEGER NTLRMS
 REAL STRLCN(10,2)
 INTEGER NSRVAR
 REAL ATRLCN(10,2)
 REAL OPMSR(10)
 REAL QTRCST(10)
 INTEGER NWRKSA(20)
 REAL CNTRD(20,2)
 REAL PTRLCN(10,2)
 INTEGER NTIMES
 INTEGER MNUVAR
  OTHER VARIABLE
 INTEGER CNTR
 REAL D(10)
 INTEGER STC(10)
  IF FIRST RUN THEN
 IF (NTIMES.EQ.1) THEN
    CALCULATE PERFECT TOOLROOM LOCATION
         CALL CPTRLN(NTLRMS,NSRVAR,NWRKSA,CNTRD,
     +                   PTRLCN)
   IF (MNUVAR.EQ.3) THEN
     DO 10 I=1,NTLRMS
```

```
      ATRLCN(I,1)=PTRLCN(I,1)
      ATRLCN(I,2)=PTRLCN(I,2)
0           CONTINUE
  END IF
 IF REPEAT ITERATION THEN
ELSE
  CHANGE LOCATIONS TO MAKE MORE OPTIMAL
   CALL CHLCN(NTLRMS,NSRVAR,ATRLCN,OPMSR,QTRCST,D,STC)
   CALL TLKUSR(NTLRMS,STRLCN,ATRLCN,PTRLCN,D,STC)
END IF
       END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC


 CREATE DISTANCE MATRIX
SUBROUTINE CREDIS(NTLRMS,NWRKRS,WRKLCN,ATRLCN,
    +        DTRWRK)
 INPUT
INTEGER NTLRMS
INTEGER NWRKRS
REAL WRKLCN(100,2)
REAL ATRLCN(10,2)
 OUTPUT
REAL DTRWRK(10,100)
 FOR EACH WORKER LOOP
   FOR EACH TOOLROOM LOOP
      CALCULATE WORKERS DISTANCE TO TOOLROOM
   END LOOP
 END LOOP
DO 200 I=1,NTLRMS
  DO 200 J=1,NWRKRS
    X2=(WRKLCN(J,1)-ATRLCN(I,1))**2
    Y2=(WRKLCN(J,2)-ATRLCN(I,2))**2
    DTRWRK(I,J)=SQRT(X2+Y2)
00       CONTINUE
         END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC


 CALCULATE AVERAGE DISTANCE FROM SERVICE AREA TO TOOLROOM
SUBROUTINE CAVDIS(NTLRMS,TLRMNM,NWRKRS,WRKLCN,NSRVAR,
    +      SRVNMS,SAPRIM,ATRLCN,DTRWRK,ADSATR,
    +      NWRKSA,CNTRD,NTIMES,MNUVAR,OPTIMES)
 INPUT
INTEGER NTLRMS
CHARACTER*20 TLRMNM(10)
INTEGER NWRKRS
REAL WRKLCN(100,2)
INTEGER NSRVAR
CHARACTER*20 SRVNMS(20)
REAL SAPRIM(20,4)
REAL ATRLCN(10,2)
REAL DTRWRK(10,100)
 OUTPUT
REAL ADSATR(20,10)
 INPUT
```

```
INTEGER NWRKSA(20)
REAL CNTRD(20,2)
INTEGER NTIMES
INTEGER MNUVAR
INTEGER OPTIMES
      INTEGER COUNT
 FOR EACH WORKER LOOP
   FOR EACH TOOLROOM LOOP
     ADD TO SERVICE AREA THAT WORKER IS IN 'S DISTANCE
 (SERVICE AREA,TOOLROOM   MATRIX)
   END LOOP
 END LOOP
 FOR EACH SERVICE AREA LOOP
   DIVIDE SUMMED DISTANCE BY THE NUMBER OF WORKERS IN THE SERVICE AREA
 END LOOP
IF ((MNUVAR.EQ.2).AND.(NTIMES.EQ.1)) THEN
          WRITE(*,10)
0         FORMAT(//,1X,'ENTER THE AVERAGE DISTANCE BETWEEN:')
  DO 100 I=1,NTLRMS
    DO 100 J=1,NSRVAR
4             WRITE(*,15)TLRMNM(I),SRVNMS(J)
5       FORMAT(1X,'TOOLROOM: ',A20,/,1X,'SERVICE AREA:',A20,/,1X,'REAL',
    +/,1X,'ENTER:')
      READ(*,20,ERR=21)ADSATR(J,I)
0       FORMAT(F8.0)
       GOTO 55
1      WRITE(*,401)
       GOTO 9
5        WRITE(*,56)
6      FORMAT(/,1X,'ENTER 1 TO CORRECT ELSE <RETURN> TO CONTINUE')
      READ(*,58,ERR=59)III
8             FORMAT(I2)
       GOTO 60
9      WRITE(*,401)
       GO TO 55
0      CONTINUE
      IF (III.EQ.1) GOTO 14
00        CONTINUE
END IF
IF (MNUVAR.EQ.3) THEN
 DO 200 I=1,NTLRMS
   DO 200 J=1,NSRVAR
     X2=(CNTRD(J,1)-ATRLCN(I,1))**2
     Y2=(CNTRD(J,2)-ATRLCN(I,2))**2
     ADSATR(J,I)=SQRT(X2 + Y2)
00       CONTINUE
END IF
00        CONTINUE
  DO 400 I=1,NTLRMS
    DO 400 J=1,NSRVAR
      IF (NWRKSA(J).NE.0) THEN
        ADSATR(J,I)=ADSATR(J,I)/NWRKSA(J)
      END IF
00        CONTINUE
END IF
01     FORMAT(,/,1X,'***ERROR***MISTYPED DATA',/)
        END


CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
 CALCULATE WEIGHTED AVERAGE DISTANCE TRAVELED TO TOOLROOM
SUBROUTINE WAVDIS(NTLRMS,NWRKRS,NSRVAR,
    +        ADSATR,WDTLRM,NWRKSA)
 INPUT
INTEGER NTLRMS
INTEGER NWRKRS
INTEGER NSRVAR
REAL ADSATR(20,10)
 OUTPUT
REAL WDTLRM(10)
 INPUT
INTEGER NWRKSA(20)
 FOR EACH TOOLROOM LOOP
   FOR EACH SERVICE AREA LOOP
     CALCULATE THE FOLLOWING FORMULA:
      (*WORKERS IN A SERVICE AREA/#WORKERS)
      * AVERAGE DISTANCE FROM SERVICE AREA TO TOOLROOM
   END LOOP
 END LOOP
DO 200 I=1,NTLRMS
  DO 200 J=1,NSRVAR
    IF (NWRKRS.NE.0) THEN
      WDTLRM(I)=(NWRKSA(J)/NWRKRS)*ADSATR(J,I)
    ELSE
      WRITE(*,185)
85      FORMAT(1X,'SUBROUTINE WAVDIS DETECTS # WORKERS IS 0')
           END IF
00     CONTINUE
        END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

    CALCULATE S
SUBROUTINE CALCS(I,NWNDWS,NATNTS,S)


 INPUT
INTEGER I
INTEGER NWNDWS(10),NATNTS(10)
 OUTPUT
INTEGER S(10)


 IF #ATTENDANTS > #WINDOWS THEN
  IF (NATNTS(I).GT.NWNDWS(I)) THEN
    S = #WINDOWS
        S(I)=NWNDWS(I)
 ELSE
  ELSE
    S = #ATTENDANTS
        S(I)=NATNTS(I)
 END IF
  END IF
END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

    DO MULTICHANNEL SINGLE PHASE QUEUING THEORY
SUBROUTINE MCHQTH(I,S,MU,LAMBDA,
    +                     PO,PN,RHO,LS,LQ,WS,WQ)

 INPUT
INTEGER I
INTEGER S(10)
REAL MU(10)
REAL LAMBDA(10)
 OUTPUT
REAL PO(10)
REAL PN(50,10)
REAL RHO(10)
REAL LS(10)
REAL LQ(10)
REAL WS(10)
REAL WQ(10)


REAL LOVRMU
INTEGER E

RHO(I) = LAMBDA(I)/(S(I)*MU(I))
A = 1 - RHO(I)
B = 1/A
LOVRMU = LAMBDA(I)/MU(I)
D = LOVRMU ** S(I)
E = IFACT(S(I))
F = (D/E)*B
SUM = 0.
DO 10 J=1,(S(I)-1)
        PN(J,I) =  (LOVRMU**J)/IFACT(J)
        SUM = SUM + PN(J,I)
0       CONTINUE
G = SUM + F
PO(I) = 1/G
IF2 = IFACT(S(I))
LISRJS=0
LIPROD=0
DO 20 J=S(I),50
        ZLJ=LOVRMU**J
        IEXP=J-S(I)
        ISRJS=S(I)**IEXP
        IPROD=IF2*ISRJS
        IF ((ISRJS.LT.LISRJS).OR.(IPROD.LT.LIPROD)) THEN
                DO 16 K=J,50
                        PN(K,I)=0.0
6                       CONTINUE
                GOTO 21
        END IF
        PN(J,I)=ZLJ/IPROD
0       CONTINUE
1       DO 30 J=1,50
        PN(J,I)=PN(J,I)*PO(I)
0       CONTINUE
 LQ
Z = PO(I) * D / E
```

```
Y = .92088019613
LQ(I) = Z * Y
LS(I) = LQ(I) + (LAMBDA(I) / MU(I))
WQ(I) = LQ(I) / LAMBDA(I)
WS(I) = WQ(I) + (1/MU(I))
        END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC


    DO SINGLE CHANNEL SINGLE PHASE QUEUING THEORY
SUBROUTINE SCHQTH(I,MU,LAMBDA,
    +                    WQ,WS,PO,PN,RHO,LS,LQ)

 INPUT
INTEGER I
REAL MU(10)
REAL LAMBDA(10)
 OUTPUT
        REAL WQ(10),WS(10),PO(10)
REAL PN(50,10)
REAL RHO(10),LS(10),LQ(10)
 OTHER VARIABLES
REAL LOVRMU

  IF (LAMBDA(I).NE.MU(I)) THEN
    LOVRMU = LAMBDA(I)/MU(I)
    PO(I) = 1 - (LOVRMU)
 PN
    DO 10 J=1,50
        PN(J,I)=PO(I)*((LOVRMU)**J)
0           CONTINUE
    RHO(I) = LAMBDA(I) / MU(I)
    LS(I) = LAMBDA(I) / (MU(I) - LAMBDA(I))
    LQ(I) = (LAMBDA(I)*LAMBDA(I))/(MU(I)*(MU(I)-LAMBDA(I)))
    WS(I) = 1 / (MU(I)- LAMBDA(I))
    WQ(I) = LAMBDA(I) / (MU(I)*(MU(I) - LAMBDA(I)))

          ELSE
 DO SOMETHING TO NOTIFY USER OF ERROR
  END IF
0       CONTINUE
END


CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC


 CALCULATE CS
SUBROUTINE CALCCS(NTLRMS,WQ,WS,ATNWGS,WRKWGS,CS)
 INPUT
INTEGER NTLRMS
REAL WQ(10),WS(10)
REAL ATNWGS(10)
REAL WRKWGS
REAL CS(10)
 CS = (WS - WQ) * (ATTENDANTS WAGES + WORKERS WAGES)
DO 100 I=1,NTLRMS
   CS(I) = (WS(I) - WQ(I)) * (ATNWGS(I)+WRKWGS)
```

```
UU      CONTINUE
        END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC


 CALCULATE CW
SUBROUTINE CALCCW(NTLRMS,WQ,WRKWGS,CW)
 INPUT
INTEGER NTLRMS
REAL WQ(10)
REAL WRKWGS
 OUTPUT
REAL CW(10)
 CW = WQ * WORKERS WAGES
DO 100 I=1,NTLRMS
   CW(I) = WQ(I) * WRKWGS
OO      CONTINUE
        END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC


 CALCULATE OPTIMUM MEAN SERVICE RATE FOR TOOLROOM
SUBROUTINE COPMSR(NTLRMS,LAMBDA,OPMSR,CS,CW)
 INPUT
INTEGER NTLRMS
REAL LAMBDA(10)
 OUTPUT
REAL OPMSR(10)
 INPUT
REAL CS(10)
REAL CW(10)
 OPMSR = LAMBDA + SQRROOT(LAMBDA*CW/CS)
DO 100 I=1,NTLRMS
   XX=CW(I)/CS(I)
   XX=LAMBDA(I)*XX
   IF (XX.GT.O.) THEN
    OPMSR(I) = LAMBDA(I) + SQRT(XX)
   ELSE
    OPMSR(I) = LAMBDA(I) + SQRT(-1*XX)
   END IF
OO      CONTINUE
        END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC


 CALCULATE QUEUING TOOLROOM COSTS
SUBROUTINE CQTRCS(NTLRMS,ANWFSA,WRKWGS,WLKRT,NSRVAR,
    +              ADSATR,WDTLRM,QTRCST,CS,CW)
 INPUT
INTEGER NTLRMS
REAL ANWFSA(10)
REAL WRKWGS
REAL WLKRT
INTEGER NSRVAR
```

```
REAL ADSATR(20,10)
REAL WDTLRM(10)
 OUTPUT
REAL QTRCST(10)
 INPUT
REAL CS(10),CW(10)
DO 10 I=1,10
  QTRCST(I)=0.
0        CONTINUE
DO 300 I=1,NTLRMS
  DO 300 J=1,NSRVAR
 CALC TOTAL DIST A WORKER WALKS
    A=2*ADSATR(J,I)
 DIVIDE BY WALK RATE TO SEE HOW MANY MINUTES IT TAKES
    B=A/WLKRT
 CALCULATE WAGE PER MINUTE
    C=WRKWGS/60.0
 COST FOR ONE WORKER
    CST41=B*C+CS(I)+CW(I)
    QTRCST(I)=QTRCST(I)+CST41*ANWFSA(I)
00       CONTINUE
         END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

 CALCULATE TOTAL TOOLROOM COST
SUBROUTINE CTTRCS(NTLRMS,BLDCST,INVCST,QTRCST,
    +            TTRCST)
 INPUT
INTEGER NTLRMS
REAL BLDCST(10),INVCST(10)
REAL QTRCST(10)
 OUTPUT
REAL TTRCST(10)
    TTRCST = QTRCST(TOOLROOM) + BLDCST(TOOLROOM) + INVCST(TOOLROOM)
DO 100 I=1,NTLRMS
   TTRCST(I) = QTRCST(I) + BLDCST(I) + INVCST(I)
00       CONTINUE
         END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

 SUM TOTAL TOOLROOM COSTS
SUBROUTINE CSTALL(NTLRMS,TTRCST,TOTCST)
 INPUT
INTEGER NTLRMS
REAL TTRCST(10)
 OUTPUT
REAL TOTCST
 FOR I=1..#TOOLROOMS LOOP
   TOTCST = TOTCST + TTRCST(I)
 END LOOP
TOTCST=0.0
DO 100 I=1,NTLRMS
  TOTCST = TOTCST + TTRCST(I)
00       CONTINUE
```

```
      END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

C     CALCULATE PERFECT TOOLROOM LOCATION
      SUBROUTINE CPTRLN(NTLRMS,NSRVAR,NWRKSA,CNTRD,
     +                  PTRLCN)
         INTEGER NTLRMS
      INTEGER NSRVAR
      INTEGER NWRKSA(20)
      REAL CNTRD(20,2)
      REAL PTRLCN(10,2)
C     OTHER VARIABLES
C      TOTAL SERVICE AREA WEIGHT
      INTEGER TSAWGT
C      SORTING ARRAY (SORTED BY WEIGHT) HEAVIES FIRST
      INTEGER SRTDWT(10)
C      NUMBER OF TOOLROOMS UN ASSIGNED
      INTEGER TLRMLFT
C     WEIGHT OF THE CURRENT PARTITION
      INTEGER PRTWGT
C     AVERAGE WEIGHT OF SERVICE AREAS
      INTEGER AWGT
C     WEIGHT THAT HAS BEEN ASSIGNED TO A TOOLROOM
      INTEGER WGTASS
C     SERVICE AREAS LEFT THAT HAVE NO TOOLROOM ACCESS YET
      INTEGER SALEFT
C     POINTERS
      INTEGER STARTI,LASTI
C     IF #SERVICE AREAS == #TOOLROOMS THEN
      IF (NSRVAR.EQ.NTLRMS) THEN
        DO 3 I=1,NSRVAR
C         ASSIGN LOCATIONS OF TOOLROOMS AS THOSE CLOSEST TO CENTROIDS
C          OF SERVICE AREAS
          PTRLCN(I,1)=CNTRD(I,1)
          PTRLCN(I,2)=CNTRD(I,2)
3               CONTINUE
C     ELSE IF #SERVICE AREAS > #TOOLROOMS
      ELSE IF (NSRVAR.GT.NTLRMS) THEN
C           ASSIGN LOCATION OF TOOLROOM BASED ON WEIGHTING CENTROIDS, I.E.
C           LOCATION FOR EACH CENTROID AND EXTRA TOOLROOMS NEAR 'HEAVIEST'
C           CENTROIDS (MOST WORKERS)

C      SIMPLE BUBBLE SORT WITH HEAVIEST COMING TO THE TOP
        DO 5 I=1,NSRVAR
          SRTDWT(I)=I
5               CONTINUE
        DO 10 I=1,(NSRVAR-1)
          DO 10 J=(I+1),NSRVAR
            IF (NWRKSA(SRTDWT(I)).LT.NWRKSA(SRTDWT(J))) THEN
              ITMP=SRTDWT(I)
              SRTDWT(I)=SRTDWT(J)
              SRTDWT(J)=ITMP
            END IF
10              CONTINUE

        TSAWGT=0
        DO 12 I=1,NSRVAR
```

```
      TSAWGT=TSAWGT+NWRKSA(I)
2          CONTINUE
   I=1
   WGTASS=0
   SALEFT=NSRVAR
   TLRMLFT=NTLRMS
5          AWGT=(TSAWGT-WGTASS)/TLRMLFT
   STARTI=I
   LASTI=I
   PRTWGT=NWRKSA(SRTDWT(I))
0          I=I+1
   IF (PRTWGT+NWRKSA(SRTDWT(I)).LT.AWGT) THEN
     PRTWGT=PRTWGT+NWRKSA(SRTDWT(I))
     LASTI=I
     GOTO 20
   ELSE IF (PRTWGT+NWRKSA(SRTDWT(I)).EQ.AWGT) THEN
       PRTWGT=PRTWGT+NWRKSA(SRTDWT(I))
   END IF
   IF (STARTI.EQ.LASTI) THEN
     PTRLCN(TLRMLFT,1)=CNTRD(SRTDWT(LASTI),1)
     PTRLCN(TLRMLFT,2)=CNTRD(SRTDWT(LASTI),2)
   ELSE
     X=0.
     Y=0.
     DO 25 K=STARTI,LASTI
       X=X+NWRKSA(SRTDWT(K))*CNTRD(SRTDWT(K),1)
       Y=Y+NWRKSA(SRTDWT(K))*CNTRD(SRTDWT(K),2)
5            CONTINUE
            PTRLCN(TLRMLFT,1)=X/PRTWGT
     PTRLCN(TLRMLFT,2)=Y/PRTWGT
   END IF
   TLRMLFT=TLRMLFT-1
   IF (TLRMLFT.GT.1) THEN
     WGTASS=WGTASS+PRTWGT
     SALEFT=SALEFT-(LASTI-STARTI+1)
     GOTO 15
   ELSE
     IF (TLRMLFT.EQ.1) THEN
       X=0.
       Y=0.
       DO 30 I=LASTI+1,NSRVAR
        X=X+NWRKSA(SRTDWT(I))*CNTRD(SRTDWT(I),1)
        Y=Y+NWRKSA(SRTDWT(I))*CNTRD(SRTDWT(I),2)
0              CONTINUE
              PTRLCN(1,1)=X/PRTWGT
       PTRLCN(1,2)=Y/PRTWGT
     END IF
   END IF
 ELSE IF #SERVICE AREAS < #TOOLROOMS
ELSE IF (NSRVAR.LT.NTLRMS) THEN
        ASSIGN TOOLROOM LOCATION BASED ON WEIGHTING CENTROIDS
   DO 50 I=1,NSRVAR
    ASSIGN LOCATIONS OF TOOLROOMS AS THOSE CLOSEST TO CENTROIDS
     OF SERVICE AREAS
     PTRLCN(I,1)=CNTRD(I,1)
     PTRLCN(I,2)=CNTRD(I,2)
0          CONTINUE
 SIMPLE BUBBLE SORT WITH HEAVIEST COMING TO THE TOP
   DO 60 I=1,NSRVAR
     SRTDWT(I)=I
```

```
0          CONTINUE
   DO 100 I=1,(NSRVAR-1)
     DO 100 J=(I+1),NSRVAR
       IF (NWRKSA(SRTDWT(I)).LT.NWRKSA(SRTDWT(J))) THEN
         ITMP=SRTDWT(I)
         SRTDWT(I)=SRTDWT(J)
         SRTDWT(J)=ITMP
       END IF
00         CONTINUE

   TSAWGT=0
   DO 120 I=1,NSRVAR
     TSAWGT=TSAWGT+NWRKSA(I)
20         CONTINUE
   I=NSRVAR+1
   WGTASS=0
   SALEFT=NSRVAR
   TLRMLFT=NTLRMS-NSRVAR
50         AWGT=(TSAWGT-WGTASS)/TLRMLFT
   STARTI=I
   LASTI=I
   PRTWGT=NWRKSA(SRTDWT(I))
00         I=I+1
   IF (PRTWGT+NWRKSA(SRTDWT(I)).LT.AWGT) THEN
     PRTWGT=PRTWGT+NWRKSA(SRTDWT(I))
     LASTI=I
     GOTO 200
   END IF
   IF (STARTI.EQ.LASTI) THEN
     PTRLCN(TLRMLFT,1)=CNTRD(SRTDWT(LASTI),1)
     PTRLCN(TLRMLFT,2)=CNTRD(SRTDWT(LASTI),2)
   ELSE
     X=0.
     Y=0.
     DO 250 K=STARTI,LASTI
       X=X+NWRKSA(SRTDWT(K))*CNTRD(SRTDWT(K),1)
       Y=Y+NWRKSA(SRTDWT(K))*CNTRD(SRTDWT(K),2)
50           CONTINUE
           PTRLCN(TLRMLFT,1)=X/PRTWGT
     PTRLCN(TLRMLFT,2)=Y/PRTWGT
   END IF
   TLRMLFT=TLRMLFT-1
   IF (TLRMLFT.GT.1) THEN
     WGTASS=WGTASS+PRTWGT
     SALEFT=SALEFT-(LASTI-STARTI+1)
     GOTO 150
   ELSE
     X=0.
     Y=0.
     DO 300 I=LASTI+1,NSRVAR
       X=X+NWRKSA(SRTDWT(I))*CNTRD(SRTDWT(I),1)
       Y=Y+NWRKSA(SRTDWT(I))*CNTRD(SRTDWT(I),2)
00           CONTINUE
           PTRLCN(1,1)=X/PRTWGT
     PTRLCN(1,2)=Y/PRTWGT
   END IF
 END IF
  ADJUST FOR INFEASIBLE AREAS
 CALL AFINA(NTLRMS,PTRLCN)
       END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

 CALCULATE FACTORIAL
INTEGER FUNCTION IFACT(N)
INTEGER N
IF ((N.EQ.0).OR.(N.EQ.1)) THEN
        IFACT = 1
ELSE
        IFACT = 1
        DO 10 I=1,N
                IFACT=IFACT*I
0               CONTINUE
END IF
RETURN
END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

 ADJUST FOR INFEASIBLE AREAS

SUBROUTINE AFINA(NTLRMS,PTRLCN)
INTEGER NTRLMS
REAL PTRLCN(10,2)
 LOCAL VARIABLES
  INFEASIBLE AREAS
REAL INFSBL(5,4)
   NUMBER OF INFEASIBLE AREAS
INTEGER NINFS
 GET NUMBER OF INFEASIBLE AREAS
        WRITE(*,10)
0       FORMAT(//,1X,'INPUT THE NUMBER OF INFEASIBLE AREAS',/,1X,
    +'(INTEGER)',/,1X,'ENTER:')
READ(*,20,ERR=21)NINFS
0       FORMAT(I3)
       GOTO 24
1      WRITE(*,901)
       GOTO 5
4       WRITE(*,25)NINFS
5       FORMAT(//,1X,'NUMBER OF INFEASIBLE AREAS IS: ',I4)
7       WRITE(*,30)
0       FORMAT(/,1X,'ENTER 1 TO CHANGE ELSE <RETURN> TO CONTINUE')
READ(*,40,ERR=41)III
0       FORMAT(I2)
       GOTO 42
1      WRITE(*,901)
       GOTO 27
2      CONTINUE
IF (III.EQ.1) GOTO 5
 FOR EACH INFEASIBLE AREA
DO 900 I=1,NINFS
 GET CORNERS OF INFEASIBLE AREAS
99        WRITE(*,200)I
00        FORMAT(//,1X,'INPUT THE FOLLOWING DATA FOR '
    +          ,' INFEASIBLE AREA #',I2)
 GET INFEASIBLE AREA PERIMETERS
04        WRITE(*,205)
05      FORMAT(5X,'X-COORD. OF LEFTMOST BOUNDARY ',/,1X,'REAL',
```

```
     +/,1X,'ENTER:')
           READ(*,210,ERR=211)INFSBL(I,1)
10       FORMAT(F8.0)
        GOTO 214
11      WRITE(*,901)
        GOTO 204
14         WRITE(*,215)
15       FORMAT(5X,'Y-COORD. OF BOTTOMMOST BOUNDARY ',/,1X,'REAL',
     +/,1X,'ENTER:')
           READ(*,220,ERR=221)INFSBL(I,2)
20       FORMAT(F8.0)
        GOTO 224
21      WRITE(*,901)
        GO TO 214
24         WRITE(*,225)
25       FORMAT(5X,'X-COORD. OF RIGHTMOST BOUNDARY ',/,1X,'REAL',
     +/,1X,'ENTER:')
           READ(*,230,ERR=231)INFSBL(I,3)
30       FORMAT(F8.0)
        GOTO 234
31      WRITE(*,901)
        GOTO 224
34         WRITE(*,235)
35       FORMAT(5X,'Y-COORD. OF TOPMOST BOUNDARY ',/,1X,'REAL',
     +/,1X,'ENTER:')
           READ(*,240,ERR=241)INFSBL(I,4)
40       FORMAT(F8.0)
        GOTO 299
41      WRITE(*,901)
        GOTO 234
99         WRITE(*,300)I
00         FORMAT(1X,'INFEASIBLE AREA #',I3,'S BOUNDARIES ARE:')
 GET SERVICE AREA PERIMETERS
           WRITE(*,301)INFSBL(I,1)
01       FORMAT(5X,'X-COORD. OF LEFTMOST BOUNDARY IS ',F8.4)
  WRITE(*,302)INFSBL(I,2)
02       FORMAT(5X,'Y-COORD. OF BOTTOMMOST BOUNDARY IS ',F8.4)
  WRITE(*,303)INFSBL(I,3)
03       FORMAT(5X,'X-COORD. OF RIGHTMOST BOUNDARY IS ',F8.4)
  WRITE(*,304)INFSBL(I,4)
04       FORMAT(5X,'Y-COORD. OF TOPMOST BOUNDARY IS ',F8.4)
35         WRITE(*,336)
36      FORMAT(/,1X,'ENTER 1 TO CHANGE ELSE <RETURN> TO CONTINUE')
  READ(*,338,ERR=339)III
38         FORMAT(I2)
        GOTO 340
39      WRITE(*,901)
        GOTO 335
40      CONTINUE
  IF (III.EQ.1) GOTO 199
   CHECK EACH TOOLROOM
DO 800 J=1,NTLRMS
  DISTX1=INFSBL(I,3)-PTRLCN(J,1)
  DISTX2=PTRLCN(J,1)-INFSBL(I,1)
  DISTY1=INFSBL(I,4)-PTRLCN(J,2)
  DISTY2=PTRLCN(J,2)-INFSBL(I,2)
        IF TOOLROOM IN INFEASIBLE AREA THEN
          RELOCATE TOOLROOM ON EDGE
        END IF
IPTRX=0
```

```
IPTRY=0
IF (DISTX2.GE.0.) THEN
  IF (DISTX1.GE.0.) THEN
    IF (DISTX1.GE.DISTX2) THEN
      XLESS=DISTX2
      IPTRX=1
    ELSE
      XLESS=DISTX1
      IPTRX=3
    END IF
    IF (DISTY1.GE.0.) THEN
      IF (DISTY2.GE.0.) THEN
        IF (DISTY1.GE.DISTY2) THEN
          YLESS=DISTY2
          IPTRY=2
        ELSE
          YLESS=DISTY1
          IPTRY=4
        END IF
        IF (YLESS.LT.XLESS) THEN
          PTRLCN(I,2)=INFSBL(J,IPTRY)
        ELSE IF (XLESS.LT.YLESS) THEN
          PTRLCN(I,1)=INFSBL(J,IPTRX)
        ELSE
          PTRLCN(I,2)=INFSBL(J,IPTRY)
          PTRLCN(I,1)=INFSBL(J,IPTRX)
        END IF
      END IF
    END IF
  END IF
END IF
00       CONTINUE
00       CONTINUE
01      FORMAT(,/,1X,'***ERROR***MISTYPED DATA',/)
END
```